

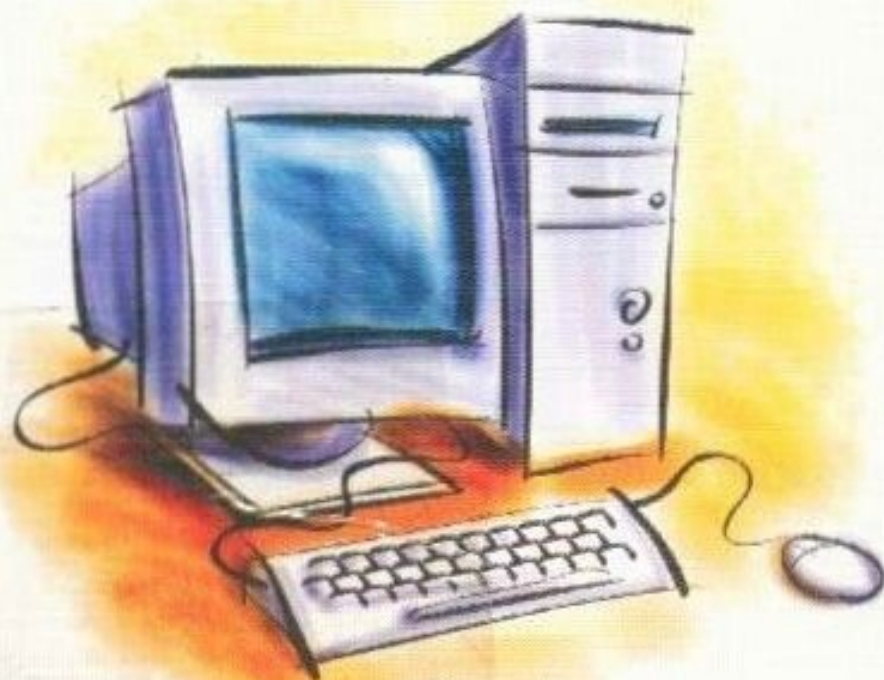
ThS. ĐẬU MẠNH HOÀN, TRẦN DŨNG

Phương pháp giải Các dạng bài tập

Tin học 11

Bổ trợ và nâng cao

Biên soạn theo chương trình & SGK
phân ban mới của bộ GD&ĐT



TT TT-TV • ĐHQGHN

004
ĐA-H
2007

LC/02187



NHÀ XUẤT BẢN ĐẠI HỌC QUỐC GIA HÀ NỘI

ThS. ĐẬU MẠNH HOÀN, TRẦN DŨNG

Phương pháp giải Các dạng bài tập **Tin học 11**

Bổ trợ và nâng cao

Biên soạn theo chương trình & SGK
phân ban mới của bộ GD&ĐT



NHÀ XUẤT BẢN ĐẠI HỌC QUỐC GIA HÀ NỘI

NHÀ XUẤT BẢN ĐẠI HỌC QUỐC GIA HÀ NỘI

16 Hàng Chuối - Hai Bà Trưng - Hà Nội

ĐT: (04) 9715013; (04) 7685236. Fax: (04) 9714899

Chịu trách nhiệm xuất bản:

Giám đốc **PHÙNG QUỐC BẢO**
Tổng biên tập **NGUYỄN BÁ THÀNH**

Biên tập nội dung

THU HIÊN

Sửa bản in

HOÀNG VĨNH

Trình bày bìa

SƠN KỲ

PHƯƠNG PHÁP GIẢI CÁC DẠNG BÀI TẬP TIN HỌC 11

Mã số: 1L-237ĐH2007

In 2.000 cuốn, khổ 16 × 24 cm tại Công ty cổ phần Văn hoá Tân Bình.

Số xuất bản: 774-2007/CXB/02-119/ĐHQG HN, ngày 24/09/2007.

Quyết định xuất bản số: 549/LK/XB

In xong và nộp lưu chiểu quý IV năm 2007.

LỜI NÓI ĐẦU

Công nghệ thông tin là ngành đi đầu trong sự phát triển của khoa học và công nghệ, làm động lực đẩy nhanh công cuộc công nghiệp hoá và hiện đại hoá của bất kì quốc gia nào. Trong chiến lược chung của sự phát triển kinh tế đất nước, những năm qua Đảng và Nhà nước ta đã chú trọng đến sự phát triển của ngành khoa học mũi nhọn này. Từ năm học 2006 - 2007, theo quyết định của Bộ Giáo dục và Đào tạo, môn Tin học đã trở thành môn học chính khóa ở bậc Trung học phổ thông.

Cũng giống các môn học khác, để học tốt môn tin học, ngoài sự đam mê và lựa chọn phương pháp học tốt thì sách tham khảo là người bạn không thể thiếu, đồng hành với những ai muốn đi đến những thành công.

Để thiết thực giúp các em có được một tài liệu tham khảo tốt, chúng tôi tổ chức biên soạn cuốn sách:

“Phương pháp giải các dạng bài tập Tin học 11”.

Cuốn sách được soạn trên cơ sở chương trình giảng dạy tin học 11 Phổ thông do Bộ Giáo dục và Đào tạo ban hành năm 2007. Nội dung được thiết kế theo từng bài học cụ thể, trong đó chia thành hai phần:

Phần 1: *Phương pháp giải toán:* Nội dung phần này giới thiệu các kiến thức trọng tâm của bài học và các hướng dẫn cụ thể về phương pháp giải các bài tập liên quan đến nội dung bài học

Phần 2: *Các bài tập áp dụng:* Nội dung phần này được biên soạn thành 3 phần: các bài tập cơ bản; các bài tập nâng cao và cuối cùng là các bài tập tự luận để các em thực hành

Ở chương cuối cùng (chương 7) trình bày một số bài toán nâng cao, các bài thực hành trên máy tính và một số đề kiểm tra tham khảo.

Chúng tôi hi vọng cuốn sách sẽ là sự hài lòng và thiết thực giúp đỡ các em trong quá trình học tập bộ môn này. Ngoài ra, sách còn có thể làm tài liệu tham khảo cho các thầy, cô giáo và các bạn đồng nghiệp.

Trong quá trình biên soạn chắc không tránh khỏi các sai sót, rất mong nhận được các ý kiến góp ý của các em học sinh và các thầy, cô giáo để chúng tôi có thể hoàn thiện hơn cho những lần tái bản sau.

Chúc các em thành công với ước mơ của mình.

Mọi góp ý xin gởi về:

- Trung tâm sách giáo dục Alpha - 225 Nguyễn Tri Phương, P.9, Q.5, Tp. HCM. ĐT: (08) 8107718, 8547464.

Email: alphabookcenter@yahoo.com

Xin trân trọng cảm ơn!

Các tác giả

Pascal – NHỮNG DẤU MỐC QUAN TRỌNG

Pascal: Hỏi có người lập trình nào không biết ngôn ngữ này? Thành công vang dội của nó bắt đầu vào những năm 1980, thời gian của cuộc cách mạng trong công nghiệp máy vi tính và giai đoạn nở rộ của lập trình có cấu trúc. Có thể nói Pascal trở thành điểm bắt đầu cho một kỷ nguyên mới của các ngôn ngữ lập trình.

Sự ra đời của Pascal

Vào đầu năm 1971, bản mô tả ngôn ngữ mới của Viện Công nghệ Liên bang Thụy Sĩ được in trong số đầu tiên của tạp chí Acta Informatica. Sự ra đời của Pascal có thể được tính từ thời điểm này.

Tác giả của nó, Giáo sư Niclaus Wirth trở nên nổi tiếng vì sự xuất hiện của Pascal. Những dự án sau này của ông chứng minh hùng hồn cho thế giới rằng chìa khoá tới các bí mật của máy tính chính là ở sự kết hợp hài hoà giữa Toán học, Công nghệ và Lập trình. Và nếu tiếp cận vấn đề một cách hợp lý thì có thể tạo ra các ngôn ngữ, hệ điều hành và ngay cả các máy tính tuyệt vời vượt các chuẩn công nghiệp, chỉ bằng sức lực của những sinh viên!

Người ta thường nói đến các điểm khác biệt của Pascal so với một ngôn ngữ khác- ngôn ngữ C. Nhưng chính Dennis Ritchie, tác giả của C đã phát biểu (1993): "Tôi khẳng định rằng Pascal rất gần với C. Hai ngôn ngữ này khác biệt về chi tiết, nhưng về cơ sở chúng là giống nhau". Khi nhìn vào các kiểu dữ liệu, cũng như các phép toán trên chúng ta có thể phát hiện ra những sự giống nhau rất lớn. Mặc dù rằng ý đồ của Wirth khi tạo ra Pascal là rất khác với ý đồ của chúng tôi khi tạo ra C. Wirth tạo ngôn ngữ để giảng dạy và do vậy tất nhiên cần đáp ứng các yêu cầu sư phạm.

Khác với C, Pascal không được tạo ra để làm ngôn ngữ lập trình hệ thống. Để đề cao tính đơn giản và hiệu quả dựa trên mức độ hiểu về khoa học lập trình thời bấy giờ, Wirth chủ tâm chấp nhận các hạn chế của ngôn ngữ, trước tiên trong các vấn đề liên quan đến thế giới bên ngoài (vào/ra và các công cụ phụ thuộc hệ thống). Mặc dù vậy, nếu nghĩ rằng Pascal là ngôn ngữ chỉ dành để giảng dạy, thì sẽ sai lầm. Hãy nghe chính ý kiến của Wirth về vấn đề này (1984): "Có người cho rằng Pascal được thiết kế như một ngôn ngữ để giảng dạy. Mặc dù điều này là đúng, nhưng việc sử dụng nó để giảng dạy không phải là mục đích duy nhất. Thực tế, tôi không tin vào sự thành công của việc áp dụng trong khi học các công cụ và phương pháp mà không thể sử dụng để giải quyết các bài toán thực tế. Theo các tiêu chuẩn ngày nay thì Pascal có những yếu điểm rõ ràng khi lập trình các hệ thống lớn, nhưng 15 năm trước, nó là thoả hiệp hợp lý giữa cái mong muốn và hiệu quả".

Các đặc điểm của Pascal và những ngôn ngữ bắt nguồn từ nó:

Ngôn ngữ Pascal được Wirth tạo ra dưới ảnh hưởng các tư tưởng của C. A. R. Hoare, đăng trong công trình: "Bản về cấu trúc dữ liệu" (Notes on Data Structuring, Academic Press, 1972). Đóng góp của nhà bác học người Anh lớn tới mức có thể gọi ông là cha đỡ đầu của Pascal.

Pascal được coi như bắt đầu của kỷ nguyên lập trình cấu trúc. Tất cả bắt đầu từ bài báo của chuyên gia người Hà Lan Dijkstra E. W. "Lập trình cấu trúc"

(SStructured Programming, 1969). Trong bài báo này ông đề xuất hạn chế các cấu trúc điều khiển chương trình chỉ ở 3 dạng: tuần tự, rẽ nhánh và lặp. Từ đó, suy ra rằng câu lệnh chuyển vô điều kiện (goto) trong các ngôn ngữ ALGOL và PPL/1 (rất phổ biến thời bấy giờ) là hoàn toàn không cần thiết. Thật sự là Wirth cũng không dám loại câu lệnh này khỏi Pascal. Nhưng điều chủ yếu nằm ở chỗ khác: Lập trình cấu trúc liên quan đến nguyên tắc "từ trên xuống dưới" (làm mịn từng bước), yêu cầu tính cấu trúc của điều khiển và dữ liệu, dựa vào sự đơn giản và cơ sở toán học mà tăng độ tin cậy của phần mềm. Tất cả những điều này đều khả thi nhờ các khả năng của Pascal.

Về ý đồ khi xây dựng Pascal, Wirth viết: "Điểm mới của Pascal là đưa ra các cấu trúc và kiểu dữ liệu phong phú, cũng giống như ALGOL đưa ra các loại cấu trúc điều khiển. Trong ALGOL chỉ có 3 kiểu dữ liệu cơ sở: các số nguyên và thực, giá trị chân lý, mảng; Pascal đã đưa thêm các kiểu dữ liệu cơ sở và còn cho khả năng xác định những kiểu cơ sở mới (kiểu liệt kê, kiểu miền con), cũng như các dạng cấu trúc dữ liệu mới: bản ghi, tập hợp, tệp, mà một số trong chúng đã có trong COBOL. Và tất nhiên, quan trọng nhất là tính hệ quy trong việc mô tả các cấu trúc và hệ quả của điều này là khả năng kết hợp và lồng các cấu trúc.

Pascal có tiếp tục tồn tại ?!

Để kết luận, xin trích dẫn lời của Dennis Ritchie tác giả ngôn ngữ C: "Pascal là một ngôn ngữ thanh lịch. Nó vẫn tiếp tục tồn tại. Nó đã khởi nguồn cho không ít ngôn ngữ đàn em và có ảnh hưởng sâu sắc đến việc thiết kế các ngôn ngữ lập trình nói chung".



Niklaus Wirth

Tiểu sử Niklaus Wirth

Giáo sư Niklaus Wirth, tác giả của ngôn ngữ Pascal, tốt nghiệp Đại học Công nghệ Liên bang Thụy Sĩ

ETH tại thành phố quê hương Zuyrich vào năm 1958. Ông nhận bằng Thạc sĩ tại trường Đại học Tổng hợp Laval ở Quebec, Canada năm 1960.

Năm 1963 tại Đại học Tổng hợp California (Mỹ) dưới sự lãnh đạo của giáo sư Garry Hasky ông thực hiện đề án mở rộng ngôn ngữ Algol-60 (ngôn ngữ Euler) và bảo vệ luận án tiến sĩ.

Trong các năm 1963-1967 ông giảng dạy tại Đại học tổng hợp Stanford Mỹ. Cũng trong thời gian này ông được mời vào nhóm chuyên gia quốc tế IFIP thiết kế Algol-68.

Năm 1967, Wirth trở về nước và giảng dạy tại Tổng hợp Zuyrich. Năm 1968 ông chuyển sang ETH, tại đây bắt đầu tham gia thiết kế Pascal. Năm 1970 chương trình dịch Pascal đầu tiên được hoàn thành.

Trong thời gian 1978-1981, Wirth lãnh đạo Dự án thiết kế ngôn ngữ Modula-2, máy tính cá nhân 16-bit Lilit dựa trên nó và hệ điều hành OS Medos.

Tất cả các chương trình, kể cả chương trình hệ thống, được thực hiện hoàn toàn trên Modula-2.

Năm 1984 do công lao to lớn trong việc phát triển các ngôn ngữ lập trình và thiết kế máy tính cá nhân Lilit, ông được giải thưởng Alan Turing -- giải thưởng cao quý nhất trong giới Tin học, về ý nghĩa tương đương với giải Nobel.

Trong thời gian 1986-1989 Wirth lãnh đạo dự án phát triển ngôn ngữ Oberon, hệ điều hành hướng đối tượng Oberon và trạm làm việc 32-bit Cēeres. Rất nhiều ý tưởng của Dự án này được các đồng nghiệp từ Phòng thí nghiệm Sun Labs sử dụng cho ngôn ngữ và công nghệ Java.

Từ 1990 ông lãnh đạo Viện các hệ thống máy tính tại ETH. Năm 1999 ông nghỉ hưu một cách xứng đáng và trở thành Giáo sư danh dự của ETH.

CÁC LỖI THƯỜNG GẶP TRONG TURBO PASCAL.

1. Out of memory:

Lỗi này xảy ra khi trình biên dịch bị hết bộ nhớ. Hãy giải quyết bằng các cách sau:

- Nếu Compile / Destination đặt tới Memory thì hãy đặt tới Disk trong môi trường khép kín IDE.

- Nếu Option / Linker / Link Buffer đặt tới Memory hãy chọn lại Disk.

Nếu những xử lý trên không có tác dụng thì chương trình của bạn có thể quá lớn để dịch với kích thước bộ nhớ hiện có. Khi đó phải chia nhỏ chương trình ra để dịch.

2. Identifier expected:

Một định danh (tên) phải xem xét lại tại điểm này, có thể bạn đang khai báo lại một từ dành riêng hay từ khoá của Turbo Pascal.

3. Unknown identifier:

Định danh hay tên này chưa khai báo, hoặc không được biết đến trong modul hiện thời, thông thường có thể đó là một tên đã khai báo mà có nhầm lẫn một vài kí tự nào đó.

4. Duplicate identifier:

Định danh hay tên đã được khai báo và lại khai báo lần hai.

5. Syntax error:

Thường là lỗi cú pháp, một nội dung nào đó sai cú pháp chẳng hạn như thiếu dấu nháy trong chuỗi hằng 'abc'.

6. String constant exceeds line:

Thiếu dấu kết thúc của một hằng xâu kí tự. Kí tự nháy trên '.

7. Unexpected end of file:

Có thể gặp lỗi này do các nguyên nhân sau:

- Tệp nguồn kết thúc trước lệnh end của phần lệnh chính. Hầu như là lệnh begin và end không đi cặp với nhau.
- Một tệp Include kết thúc ở giữa phần lệnh. Mọi phần lệnh phải chứa đựng trong một tệp.
- Bạn chưa đóng chú thích.

8. Line too long:

Dòng quá dài, độ dài tối đa của dòng là 127 kí tự. Lỗi này có thể do chưa đóng dấu kết thúc ' trong một hằng chuỗi.

9. Type identifier expected:

Định danh hay tên không biểu hiện một kiểu mà đáng ra nó phải có.

10. Invalid file name:

Tên tệp không đúng hoặc chỉ định một đường dẫn không tồn tại.

11. File not found:

Trình biên dịch không tìm thấy tệp ở trong thư mục hiện hành hoặc trong bất kì đường dẫn nào đã chỉ ra.

12. Disk full:

Đĩa đã đầy. Hãy tạo không gian trống của đĩa để thực hiện.

13. Variable identifier expected:

Định danh hay tên không thể hiện một biến mà đáng ra nó phải có.

14. Error in type:

Kí hiệu này không thể bắt đầu một định nghĩa kiểu.

15. Invalid string leng:

Độ dài lớn nhất khai báo của một chuỗi phải nằm trong khoảng 1..255

16. Type mismatch:

Lỗi này xảy ra trong các trường hợp sau:

- Kiểu của biến và biểu thức không tương thích trong lệnh gán.
- Kiểu không tương thích giữa tham số thực và tham số hình thức.
- Kiểu của biểu thức không tương thích với kiểu chỉ số khi làm chỉ số mảng.
- Không tương thích giữa các kiểu của các toán hạng trong một biểu thức.

17. Lower bound greater than upper bound:

Khai báo của một kiểu miền con chỉ ra rằng cận dưới cao hơn cận trên

18. Ordinal type expected:

Kiểu thực, kiểu chuỗi, kiểu cấu trúc, kiểu con trỏ không được sử dụng trong trường hợp này. Các kiểu này không phải là các kiểu đếm được.

19. Integer constant expected:

Không viết đúng hằng số nguyên.

20. Integer or real constant expected:

Viết không đúng hằng thực và nguyên.

21. Invalid function result type:

Kiểu kết quả của hàm không hợp lệ. Các kiểu trả về của hàm hợp lệ là các kiểu đơn giản, kiểu chuỗi và kiểu con trỏ.

22. Label identifier expected:

Tên nhãn không đúng.

23. Begin expected:

Thiếu từ khoá begin ở đây hoặc có lỗi trong một khối unit hoặc chương trình.

24. End expected:

Thiếu từ khoá end ở đây hoặc có lỗi trong một khối unit hoặc chương trình, hoặc thừa begin.

25. Integer expression expected:

Biểu thức kiểu nguyên không đúng.

26. Legal assignment:

- Không thể gán giá trị cho file và biến không kiểu.
- Tên hàm chỉ có thể gán giá trị trong phần lệnh của hàm.

27. File identifier expected:

Định danh hay tên không thể hiện là một trường trong bản ghi tương ứng hoặc trong biến object.

28. Do expected:

Thiếu từ khoá Do hoặc đặt không đúng chỗ.

29. Of expected:

Thiếu từ khoá of. Từ dành riêng of không xuất hiện ở nơi nó cần có.

30. Then expected:

Thiếu từ khoá Then. Từ dành riêng Then không xuất hiện ở nơi nó cần có.

31. To or Downto expected:

Từ dành riêng To hay Downto không xuất hiện ở nơi cần có.

32. Invalid typecast:

- Kích thước của biến và kiểu nhận khác nhau.

- Bạn đang đưa một biểu thức vào nơi chỉ cho phép biến có mặt.

33. Division by zero:

Phép toán chia cho 0.

34. Invalid file type:

Thủ tục quản lí file không chấp nhận kiểu file này. Ví dụ: readln không thể dùng với file có kiểu.

35. Cannot read and write variables of this type:

Thủ tục Read, Readln, Write, Writeln không thể thực hiện với biến kiểu này.

36. String variable expected:

Biến phía trước phải là kiểu chuỗi.

37. String expression expected:

Biểu thức phía trước phải là kiểu chuỗi.

38. Constant and case types do not match:

Kiểu của hằng case không tương thích với biểu thức chọn của lệnh case.

39. Record or object variable expected:

Biến phía trước phải là kiểu bản ghi hay object.

40. Constant out of range:

- Đánh chỉ số mảng với một hằng vượt ra ngoài phạm vi.
- Gán một hằng vượt ra ngoài phạm vi cho một biến.
- Cho một hằng vượt phạm vi làm tham số cho một biến hoặc một hàm.

41. File variable expected:

Biến phía trước phải là kiểu file.

42. Integer or real expected:

Biến phía trước phải là kiểu nguyên hay kiểu thực.

43. Unit expected:

Từ khoá unit không xuất hiện ở nơi cần có.

44. “;” expected:

Thiếu dấu “;”

45. “:” expected:

Thiếu dấu “:”

46. “,” expected:

Thiếu dấu “,”

47. “(” expected:

Thiếu dấu “(”

48. “)” expected:

Thiếu dấu “)”

49. “=” expected:

Thiếu dấu “=”

50. “:=” expected:

Thiếu dấu “=”

51. “[” or “(” expected:

Thiếu dấu “(” hoặc “[”

52. “]” or “)” expected:

Thiếu dấu “]” hoặc “)”

53. “.” expected:

Thiếu dấu “.”

54. Invalid For control variable:

Biến điều khiển câu lệnh for phải là kiểu đơn giản được định nghĩa trong phần khai báo của chương trình con hiện thời.

55. Integer variable expected:

Biến phía trước phải là kiểu nguyên.

56. String length mismatch:

Độ dài của một hằng xâu kí tự không phù hợp với số thành phần trong mảng kí tự.

57. String constant expected:

Chuỗi hằng không xuất hiện ở nơi cần có.

58. Error in statement:

Kí hiệu này không thể bắt đầu một câu lệnh

59. Invalid qualifier:

Một trong các lỗi:

- Đánh chỉ số một biến không phải là mảng.
- Chỉ ra một trường trong một biến không phải là bản ghi.
- Lấy địa chỉ trả về của một biến không phải là con trỏ.

60. File access denied:

File không thể mở hoặc tạo ra.

61. Path not found:

Không tìm thấy file hay thư mục hiện tại.

62. Disk read error:

Lỗi khi đọc đĩa

63. Disk write error

Lỗi khi ghi đĩa

64. File not open:

File chưa mở (lỗi có liên quan đến các thủ tục của file)

Chương 1

MỘT SỐ KHÁI NIỆM VỀ LẬP TRÌNH VÀ NGÔN NGỮ LẬP TRÌNH

Nội dung chủ yếu của chương đó là nghiên cứu các khái niệm như: Phân loại ngôn ngữ lập trình; Chương trình dịch; Các thành phần của ngôn ngữ lập trình; Các thành phần cơ bản của ngôn ngữ lập trình Pascal.

Bài 1

KHÁI NIỆM LẬP TRÌNH VÀ NGÔN NGỮ LẬP TRÌNH

A. PHƯƠNG PHÁP GIẢI TOÁN

+ Mọi bài toán có giải thuật đều có thể giải được trên máy tính điện tử. Lập trình là việc sử dụng cấu trúc dữ liệu và các cấu trúc lệnh của một ngôn ngữ lập trình cụ thể để mô tả dữ liệu và diễn đạt các thao tác của thuật toán.

+ Ngôn ngữ lập trình là một phần mềm dùng để diễn đạt thuật toán thành một chương trình giúp cho máy tính hiểu được thuật toán đó. Ngôn ngữ lập trình bao gồm: Ngôn ngữ máy, hợp ngữ và ngôn ngữ bậc cao. Việc mô tả càng tổng quát bao nhiêu thì ngôn ngữ lập trình càng gần với ngôn ngữ tự nhiên bấy nhiêu. Các ngôn ngữ bậc cao gần với ngôn ngữ tự nhiên nên rất tiện lợi cho việc mô tả các thao tác.

+ Chương trình dịch là chương trình có chức năng chuyển đổi chương trình được viết bằng ngôn ngữ lập trình bậc cao (chương trình nguồn) thành một chương trình mà máy tính có thể hiểu và thực hiện được. Cần phải có một chương trình dịch để chuyển chương trình nguồn viết bằng một ngôn ngữ lập trình sang ngôn ngữ mã máy. Như vậy, đầu vào của chương trình dịch là chương trình nguồn và đầu ra là chương trình đích có dạng mã máy.

+ Biên dịch (compiler) là quá trình kiểm tra và phát hiện lỗi, kiểm tra tính đúng đắn của các câu lệnh đồng thời dịch toàn bộ chương trình nguồn thành một chương trình đích có thể thực hiện trên máy và có thể lưu trữ để sử dụng lại.

+ Thông dịch (interpreter) được thực hiện bằng cách lặp lại các thao tác kiểm tra tính đúng đắn của câu lệnh, chuyển đổi thành ngôn ngữ máy và tiến hành thực hiện lệnh vừa chuyển đổi.

Các chương trình dịch không đơn thuần chỉ dịch chương trình mà còn cung cấp cho người sử dụng một loạt các dịch vụ liên quan như biên soạn, lưu trữ, tìm kiếm chương trình nguồn, hỗ trợ việc hiệu chỉnh: xác định lỗi nếu có, chỉ rõ lỗi ở câu lệnh nào, cho xem các kết quả trung gian.... Vì vậy, thực chất chương trình dịch là một hệ thống phục vụ lập trình.

Mỗi ngôn ngữ thường có nhiều chương trình dịch. Ví dụ, ngôn ngữ PASCAL có các chương trình dịch TURBO PASCAL 3.3, TURBO PASCAL 5.5, TURBO PASCAL 6.0, TURBO PASCAL 7.0, FREE PASCAL 1.2, VISUAL PASCAL 2.1, ETH PASCAL... Các chương trình dịch khác nhau khác biệt ở các loại dịch vụ mà nó có thể cung cấp, đặc biệt là các dịch vụ mở rộng nâng cấp tăng cường các khả năng mới cho ngôn ngữ lập trình.

☛ Các bước để giải một bài toán:

- + Xác định bài toán.
- + Xây dựng giải thuật.
- + Lập trình.

B. BÀI TẬP ÁP DỤNG

Bài tập trắc nghiệm

Hãy chọn phương án trả lời đúng nhất.

Phần cơ bản

1.1. Bài toán có giải thuật là bài toán:

- A. Có lời giải và có kết quả sau một số bước hữu hạn.
- B. Có kết quả.
- C. Có các bước giải.
- D. Có lời giải hay.

1.2. Lập trình là:

- A. Sử dụng giải thuật để giải các bài toán.
- B. Dùng máy tính để giải các bài toán.
- C. Sử dụng cấu trúc dữ liệu và các câu lệnh của ngôn ngữ lập trình cụ thể để giải các bài toán trên máy tính.
- D. Sử dụng ngôn ngữ Pascal.

1.3. Trong một ngôn ngữ lập trình chương trình dịch là:

- A. Chương trình dịch từ tiếng Việt qua tiếng Anh và ngược lại.
- B. Chương trình Pascal.
- C. Chương trình nguồn.

D. Chương trình có chức năng chuyển đổi chương trình viết bằng ngôn ngữ lập trình qua chương trình thực hiện trên ngôn ngữ máy tính.

1.4. Đối với một ngôn ngữ lập trình có mấy kỹ thuật dịch?

- A. 1 loại (Biên dịch).
- B. 2 loại (Thông dịch và biên dịch).
- C. 2 loại (Thông dịch và hợp dịch).
- D. 2 loại (Hợp dịch và biên dịch).

1.5. Thông dịch là quá trình được thực hiện bằng cách lặp lại các bước sau:

A. - Kiểm tra tính đúng đắn của câu lệnh tiếp theo trong chương trình nguồn.

- Chuyển đổi câu lệnh đó thành một hay nhiều câu lệnh tương ứng trong ngôn ngữ máy.

- Thực hiện các câu lệnh vừa chuyển đổi được.

B. - Chuyển đổi câu lệnh đó thành một hay nhiều câu lệnh tương ứng trong ngôn ngữ máy.

- Kiểm tra tính đúng đắn của câu lệnh tiếp theo trong chương trình nguồn.

- Thực hiện các câu lệnh vừa chuyển đổi được.

C. - Thực hiện các câu lệnh vừa chuyển đổi được.

- Chuyển đổi câu lệnh đó thành một hay nhiều câu lệnh tương ứng trong ngôn ngữ máy.

- Kiểm tra tính đúng đắn của câu lệnh tiếp theo trong chương trình nguồn.

D. - Chuyển đổi câu lệnh đó thành một hay nhiều câu lệnh tương ứng trong ngôn ngữ máy.

- Kiểm tra tính đúng đắn của câu lệnh tiếp theo trong chương trình nguồn.

Phần nâng cao

1.6. Biên dịch là quá trình thực hiện qua các bước sau:

A. Dịch toàn bộ chương trình nguồn thành một chương trình đích có thể thực hiện trên máy và có thể lưu trữ để sử dụng lại khi cần thiết.

B. Duyệt, phát hiện lỗi, kiểm tra tính đúng đắn của các câu lệnh trong chương trình nguồn.

C. Dịch toàn bộ chương trình nguồn thành một chương trình đích có thể thực hiện trên máy và có thể lưu trữ để sử dụng lại khi cần thiết.

- Duyệt, phát hiện lỗi, kiểm tra tính đúng đắn của các câu lệnh trong chương trình nguồn.

D. - Duyệt, phát hiện lỗi, kiểm tra tính đúng đắn của các câu lệnh trong chương trình nguồn.

- Dịch toàn bộ chương trình nguồn thành một chương trình đích có thể thực hiện trên máy và có thể lưu trữ để sử dụng lại khi cần thiết.

- 1.7. Trong các phần mềm sau, phần mềm nào là một ngôn ngữ lập trình:
- A. MS Word.
 - B. MS Excel.
 - C. Turbo Pascal.
 - D. MS PowerPoint.
- 1.8. Trong một ngôn ngữ lập trình có các chức năng sau:
- A. Biên soạn.
 - B. Lưu trữ.
 - C. Tìm kiếm.
 - D. Có tất cả các chức năng nói trên.
- 1.9. Quy trình để dịch một chương trình ra dạng ngôn ngữ máy là:
- A. Chương trình dịch → Chương trình đích → Chương trình nguồn.
 - B. Chương trình nguồn → Chương trình dịch → Chương trình đích.
 - C. Chương trình dịch → Chương trình đích → Chương trình mã máy.
 - D. Chương trình đích → Chương trình nguồn → Chương trình mã máy.
- 1.10. Ngôn ngữ lập trình Pascal là:
- A. Ngôn ngữ lập trình bậc thấp.
 - B. Ngôn ngữ máy tính.
 - C. Ngôn ngữ lập trình bậc cao.
 - D. Ngôn ngữ biên dịch.
- 1.11. Chương trình viết bằng ngôn ngữ bậc cao có đặc điểm:
- A. Tốc độ thực hiện nhanh hơn so với chương trình được viết bằng ngôn ngữ máy.
 - B. Viết dài và mất nhiều thời gian hơn so với chương trình viết bằng ngôn ngữ máy.
 - C. Khai thác được tối đa các khả năng của máy.
 - D. Ngắn gọn, dễ hiểu, dễ hiệu chỉnh và nâng cấp, không phụ thuộc vào loại máy.
- 1.12. Chương trình viết bằng hợp ngữ có đặc điểm:
- A. Máy tính có thể hiểu được trực tiếp chương trình này.
 - B. Kiểu dữ liệu và cách tổ chức dữ liệu đa dạng, thuận tiện cho việc mô tả thuật toán.
 - C. Diễn đạt gần với ngôn ngữ tự nhiên.
 - D. Tốc độ thực hiện nhanh hơn so với chương trình được viết bằng ngôn ngữ bậc cao.

Bài 2

CÁC THÀNH PHẦN CỦA NGÔN NGỮ LẬP TRÌNH

A. PHƯƠNG PHÁP GIẢI TOÁN

1. Thành phần của ngôn ngữ lập trình.

Trong ngôn ngữ lập trình, các thành phần chính bao gồm: Bảng chữ cái, cú pháp và ngữ nghĩa.

+ Bảng chữ cái là tập các kí tự được dùng để viết chương trình. Không được phép dùng bất kì kí tự nào ngoài các kí tự quy định trong bảng chữ cái.

+ Cú pháp là bộ quy tắc để viết chương trình. Các quy tắc này đảm bảo tính đơn nhất khi dịch chương trình nguồn ra ngôn ngữ máy.

+ Ngữ nghĩa xác định ý nghĩa thao tác cần phải thực hiện, ứng với mỗi tổ hợp kí tự dựa vào ngữ cảnh của nó.

Như vậy, cú pháp cho biết cách viết một chương trình hợp lệ, còn ngữ nghĩa xác định tính chất, thuộc tính của các tổ hợp kí tự trong chương trình.

Các lỗi cú pháp được chương trình dịch phát hiện và thông báo cho người lập trình biết. Chỉ có các chương trình không còn lỗi cú pháp mới có thể được dịch sang ngôn ngữ máy.

Các lỗi ngữ nghĩa khó phát hiện hơn. Phần lớn các lỗi ngữ nghĩa chỉ có thể phát hiện khi thực hiện chương trình trên dữ liệu cụ thể.

2. Quy tắc đặt tên của ngôn ngữ lập trình.

Mọi đối tượng trong chương trình đều phải được đặt tên theo một quy tắc của ngôn ngữ lập trình và từng chương trình dịch cụ thể.

+ Tên dành riêng: là những tên được ngôn ngữ lập trình quy định dùng với ý nghĩa xác định (còn được gọi là từ khoá), người lập trình không được dùng với ý nghĩa khác.

Một số từ khoá:

ABSOLUTE	EXTERNAL	MOD	SHR
AND	FILE	NIL	STRING
ARRAY	FOR	NOT	THEN
BEGIN	FORWARD	OBJECT	TO
CASE	FUNCTION	OF	TYPE
CONST	GOTO	OR	UNIT
CONSTRUCTOR	IF	PACKED	UNTIL
DESTRUCTOR	IMPLEMENTATION	PROCEDURE	USES
DIV	IN	PROGRAM	VAR
DO	INLINE	RECORD	VIRTUAL
DOWNTO	INTERFACE	REPEAT	WHILE
ELSE	INTERRUPT	SET	WITH
END	LABEL	SHL	XOR

+ Tên chuẩn: là những tên được ngôn ngữ lập trình quy định dùng với một ý nghĩa nào đó, người lập trình có thể định nghĩa lại để dùng với ý nghĩa khác.

Ví dụ. Một số tên chuẩn :

ABS	INTEGER
REAL	
SQR	LONGINT
EXTENDED	
SQRT	BYTE
BREAK	

+ Tên do người lập trình đặt: là tên được dùng theo ý nghĩa riêng của từng người lập trình, tên này được khai báo trước khi sử dụng. Các tên không được trùng với tên dành riêng.

Ví dụ. Tên do người lập trình đặt :

A1
DELTA
CT_Vidu.

3. Hằng thường dùng trong ngôn ngữ lập trình.

Hằng là đại lượng có giá trị không đổi trong quá trình thực hiện chương trình. Có ba loại hằng thường dùng: Hằng số học, hằng xâu và hằng logic.

+ Hằng số học là các số nguyên và số thực, có dấu hoặc không dấu.

+ Hằng xâu là một chuỗi kí tự bất kì. Khi viết, chuỗi kí tự này được đặt trong cặp dấu nháy đơn.

+ Hằng logic là giá trị đúng (true) hoặc sai (false).

Ví dụ

+ Hằng số học :	2	0	-5	+18
	1.5	-22.36	+3.14159	0.5
	-2.236E01		1.0E-6	
+ Hằng logic:	TRUE	FALSE		
+ Hằng xâu :	'TIN HOC'			

4. Biến thường dùng trong ngôn ngữ lập trình.

Biến là đại lượng được đặt tên dùng để lưu trữ giá trị và giá trị này có thể được thay đổi trong quá trình thực hiện chương trình. Các biến dùng trong chương trình đều phải được khai báo.

5. Chú thích.

Chú thích được đặt giữa cặp dấu { } hoặc (* *) dùng để giải thích cho chương trình rõ ràng và dễ hiểu. Khi viết chương trình nên sử dụng chú thích để làm cho chương trình được rõ ràng và dễ hiểu hơn.

B. BÀI TẬP ÁP DỤNG

I. Bài tập trắc nghiệm

Hãy chọn phương án trả lời đúng nhất.

Phần cơ bản

2.1. Một ngôn ngữ lập trình có những thành phần cơ bản nào?

- A. Cú pháp và bảng chữ cái.
- B. Cú pháp và ngữ nghĩa.
- C. Bảng chữ cái, cú pháp và ngữ nghĩa.
- D. Bảng chữ cái và ngữ nghĩa.

2.2. Trong một ngôn ngữ lập trình, cú pháp dùng để?

- A. Biên soạn chương trình.
- B. Biên dịch chương trình.
- C. Xác định các thao tác thực hiện.
- D. Làm quy tắc viết chương trình.

2.3. Trong một ngôn ngữ lập trình, ngữ nghĩa dùng để?

- A. Xác định ý nghĩa thao tác cần phải thực hiện, ứng với tổ hợp kí tự dựa vào ngữ cảnh của nó.
- B. Xác định các câu lệnh của ngôn ngữ lập trình.
- C. Phát hiện lỗi cú pháp.
- D. Giải thích cú pháp các câu lệnh.

2.4. Trong ngôn ngữ lập trình, tên là:

- A. Cách gọi của các giá trị.
- B. Một dãy liên tiếp các kí tự có độ dài không quá 255 kí tự bao gồm các chữ số, chữ cái hoặc dấu gạch dưới và bắt đầu bằng chữ cái hoặc dấu gạch dưới.
- C. Một dãy liên tiếp các kí tự có độ dài không quá 127 kí tự bao gồm các chữ số.
- D. Một dãy liên tiếp các kí tự có độ dài bất kì bao gồm các chữ số, chữ cái hoặc dấu gạch dưới và bắt đầu bằng chữ cái hoặc dấu gạch dưới.

2.5. Trong ngôn ngữ lập trình, Tên dành riêng là:

- A. Một số tên gọi thông dụng.
- B. Một số tên gọi được ngôn ngữ lập trình quy định dùng với ý nghĩa xác định, người lập trình không được sử dụng với ý nghĩa khác và gọi là từ khoá.

Chương 2.

CHƯƠNG TRÌNH ĐƠN GIẢN

Nội dung của chương này nghiên cứu về: Cấu trúc chung của một chương trình; Một số kiểu dữ liệu chuẩn: Kiểu số nguyên, số thực, kí tự, logic; Phép toán, biểu thức số học, biểu thức quan hệ, biểu thức logic, hàm số học. Ngoài ra chương còn giới thiệu một số vấn đề liên quan đến phương pháp: Khai báo biến, lệnh gán, tổ chức vào/ra dữ liệu đơn giản và cách soạn thảo, dịch, thực hiện và hiệu chỉnh chương trình.

Bài 3

CẤU TRÚC CHƯƠNG TRÌNH

A. PHƯƠNG PHÁP GIẢI TOÁN

1. Cấu trúc tổng quát của một chương trình.

Một chương trình viết bằng ngôn ngữ lập trình bậc cao gồm hai phần cơ bản sau:

[<Phần khai báo>]

<Phần thân>

Trong đó [<Phần khai báo>] có thể có hoặc không và nếu có thì có thể gồm nhiều nội dung khai báo khác nhau, mỗi loại có một cú pháp riêng biệt. Phần <Phần thân> bắt buộc phải có.

2. Phần khai báo.

+ Các câu hỏi và bài tập trong phần này xoay quanh các vấn đề như: Cấu trúc chương trình (phần khai báo và phần thân). Phần khai báo bao gồm khai báo tên chương trình, khai báo thư viện sử dụng, khai báo hằng, khai báo biến và khai báo chương trình con.

+ Cú pháp để khai báo tên chương trình:

Program <tên chương trình>; Trong đó tên chương trình là tên do người sử dụng đặt theo quy định.

+ Cú pháp để khai báo thư viện: Uses <tên thư viện>; Trong đó các thư viện phải là thư viện chuẩn của ngôn ngữ hoặc là phải định nghĩa trước.

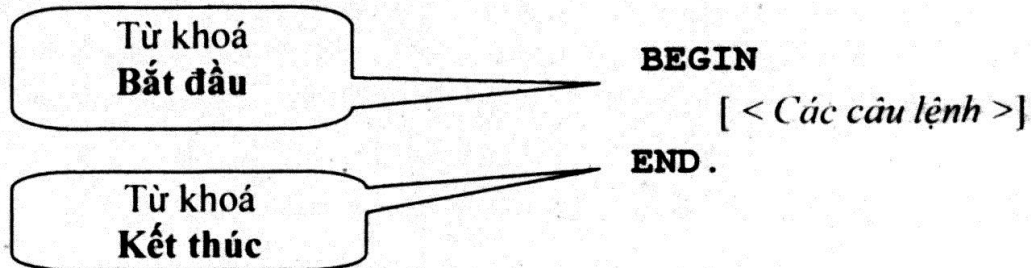
+ Cú pháp để khai báo hằng: Const <tên hằng> = <giá trị>;

+ Khai báo biến: Muốn sử dụng các biến trong chương trình thì phải khai báo theo cú pháp của nó.

3. Phần thân chương trình.

Là nơi chứa các cấu trúc và câu lệnh, thân chương trình bắt đầu bởi từ khoá **Begin** và kết thúc bằng từ khoá **End**, giữa hai từ khoá trên là các câu lệnh, giữa các câu lệnh cách nhau bởi dấu chấm phẩy (;).

Cấu trúc của phần thân chương trình như sau:



Lưu ý: sau từ khoá **END** phải có dấu chấm.

Như vậy, chương trình TP đơn giản nhất có dạng : **BEGIN** **END.**

B. BÀI TẬP ÁP DỤNG

I. Bài tập trắc nghiệm.

Hãy chọn phương án trả lời đúng nhất.

Phần cơ bản

3.1. Cấu trúc tổng quát của một chương trình gồm:

- A. Phần khai báo biến và các câu lệnh.
- B. Khai báo hằng và khai báo biến.
- C. Phần khai báo và phần thân chương trình.
- D. Phần thân chương trình và các chú thích.

3.2. Trong các phát biểu sau phát biểu nào sai?

- A. Trong một chương trình phần khai báo có thể có hoặc không.
- B. Trong một chương trình phần thân chương trình có thể có hoặc không.
- C. Trong một chương trình phần thân chương trình bắt buộc phải có.
- D. Trong một chương trình ít nhất phải có một phần là phần thân chương trình.

3.3. Cú pháp để khai báo một tiêu đề một chương trình Pascal là:

- A. Program;
- B. <Tên chương trình>;
- C. Program <Tên chương trình>;
- D. Không có đáp án nào đúng.

3.4. Cú pháp để khai báo sử dụng thư viện trong chương trình Pascal là:

- A. Uses <stdio.h>.

- B. Uses < tên thư viện>.
- C. Uses.
- D. Program <Tên thư viện>.

3.5. Cú pháp để khai báo hằng trong chương trình Pascal là:

- A. Const <tên hằng>;
- B. Const <Giá trị của hằng>;
- C. Const <tên hằng> = <Giá trị>;
- D. Const <tên hằng>:= <Giá trị>;

Phần nâng cao

3.6. Trong ngôn ngữ lập trình Pascal, phần thân chương trình được khai báo bởi cặp từ khoá:

- A. Begin
 <dãy các lệnh>;
 End;
- B. Begin
 <dãy các lệnh>;
 End,
- C. BEGIN
 < dãy các lệnh>;
 END.

D. Đáp án B và C đều đúng.

3.7. Trong các khai báo sau khai báo nào đúng?

- A. Program
- B. Program ví dụ 1;
- C. Program baitap1;
- D. Program Chuongtrinh

3.8. Trong các khai báo sau khai báo nào đúng?

- A. Uses CRT;
- B. Usses crt;
- C. Uses RCT;
- D. CRT USES;

3.9. Trong các khai báo sau khai báo nào đúng?

- A. Const a;
- B. Const pi=3.14;
- C. Const 3.14;
- D. Const hoten:= 'Nguyen Van An';

3.10. Trong các khai báo sau khai báo nào sai?

- A. Program bai1; Uses Crt; Const 3.14;
- B. Const giatri = 123.56; Program baihocdauien;
- C. Program Vi_du; Const pi = 3.14; USES CRT;
- D. Program VD; Const a=12; Const = '123';

3.11. Trong ngôn ngữ lập trình Pascal, hằng và biến khác nhau cơ bản như thế nào?

- A. Hằng không cần khai báo còn biến phải khai báo.
- B. Hằng phải khai báo còn biến không phải khai báo.
- C. Hằng là đại lượng mà giá trị không thể thay đổi được trong quá trình thực hiện chương trình, biến là đại lượng mà giá trị có thể thay đổi được trong chương trình.
- D. Hằng là đại lượng có thể thay đổi còn biến không thể thay đổi được trong quá trình thực hiện chương trình.

II. Phần tự luận.

3.1. Viết khai báo tên chương trình cho một chương trình Pascal có tên là: Bài tập 1.

3.2. Viết khai báo sử dụng thư viện CRT.

3.3. Viết hai khai báo hằng logic, 3 khai báo hằng số và 4 khai báo hằng chuỗi.

3.4. Viết cấu trúc của một chương trình Pascal gồm các phần sau:

- Khai báo tiêu đề tên chương trình.
- Khai báo sử dụng thư viện.
- Khai báo hằng.
- Thân chương trình.

3.5. Dựa vào ví dụ trong sách giáo khoa viết chương trình hiển thị dòng chữ: Turbo Pascal.

Bài 4

MỘT SỐ KIỂU DỮ LIỆU CHUẨN

A. PHƯƠNG PHÁP GIẢI TOÁN

Trong toán học, để thực hiện được tính toán ta cần phải có các tập số. Cũng tương tự như vậy, trong ngôn ngữ lập trình Pascal, để lập trình giải quyết các bài toán, cần có các tập hợp, mỗi tập hợp có một giới hạn nhất định. Ta có thể hiểu điều đó như sau: Kiểu dữ liệu chuẩn là một tập hữu hạn các giá trị, mỗi kiểu dữ liệu cần một dung lượng bộ nhớ cần thiết để lưu trữ và xác định các phép toán có thể tác động lên dữ liệu.

+ Kiểu số nguyên:

Kiểu	Phạm vi giá trị	Bộ nhớ lưu trữ một giá trị
Byte:	0 đến 255	1 byte
Word:	0 đến 65535	2 byte
Integer	-32768 đến 32767	2 byte
Shortint	-128 đến 127	1 byte
Longint	-2147483648 đến 2147483647	4 byte

+ Kiểu số thực:

Kiểu	Phạm vi giá trị	Bộ nhớ lưu trữ một giá trị
Real	$2.9 \cdot 10^{-39}$ đến $1.7 \cdot 10^{38}$	6 byte
Double	$5.0 \cdot 10^{-324}$ đến $1.7 \cdot 10^{308}$	8 byte

+ Kiểu kí tự:

Kiểu	Phạm vi giá trị	Bộ nhớ lưu trữ một giá trị
Char	256 kí tự trong bảng mã ASCII	1 byte

+ Kiểu logic:

Kiểu	Phạm vi giá trị	Bộ nhớ lưu trữ một giá trị
Boolean	True hoặc False	1 byte

Ngoài ra còn có kiểu miền con:

Kiểu miền con có dạng :

<Giá trị đầu> .. <Giá trị cuối>

Trong đó :

- *Giá trị đầu* và *Giá trị cuối* thường dùng là các hằng nguyên hoặc kí tự.
- *Giá trị đầu* nhỏ hơn hoặc bằng *Giá trị cuối*.

Kiểu miền con xác định tập các giá trị liên tiếp nhau từ *Giá trị đầu* đến *Giá trị cuối*.

Ví dụ

1..10 : Xác định tập 10 giá trị nguyên từ 1 đến 10;

'A'..'C' : Xác định tập 3 chữ cái in hoa A, B, C.

B. BÀI TẬP ÁP DỤNG

I. Bài tập trắc nghiệm

Hãy chọn phương án trả lời đúng nhất.

Phần cơ bản

4.1. Phát biểu nào dưới đây là đúng?

- A. Kiểu dữ liệu chuẩn là kiểu dữ liệu do người sử dụng tạo ra.
- B. Kiểu dữ liệu chuẩn là kiểu dữ liệu tuân theo một cú pháp chuẩn.
- C. Kiểu dữ liệu chuẩn là kiểu dữ liệu được xây dựng sẵn.
- D. Tất cả các câu đều đúng.

4.2. Kiểu dữ liệu Integer có phạm vi giá trị là:

- A. Từ 2^{15} đến 2^{16}
- B. Từ -2^{15} đến 2^{16}
- C. Từ -2^{15} đến $2^{15}-1$
- D. Từ -2^{15} đến $2^{16}-1$

4.3. Kiểu dữ liệu Byte có phạm vi giá trị là:

- A. Từ 1 đến 255
- B. Từ 0 đến 255
- C. Từ 1 đến 256
- D. Từ 0 đến 256

4.4. Kiểu dữ liệu Word có phạm vi giá trị là:

- A. Từ 1 đến $2^{16}-1$
- B. Từ 0 đến $2^{15}-1$
- C. Từ 1 đến 2^{16}
- D. Từ 0 đến $2^{16}-1$

4.5. Kiểu dữ liệu Longint có phạm vi giá trị là:

- A. Từ -2^{31} đến $2^{31}-1$
- B. Từ -2^{31} đến 2^{31}
- C. Từ 2^{31} đến $2^{31}-1$
- D. Từ -2^{31} đến $2^{31}+1$

4.6. Đối với kiểu dữ liệu Integer bộ nhớ lưu trữ một giá trị là:

- A. 1 Byte.
- B. 2 Byte.
- C. 3 Byte.
- D. 2 Byte.

4.7. Đối với kiểu dữ liệu Byte bộ nhớ lưu trữ một giá trị là:

- A. 1 Byte.
- B. 2 Byte.
- C. 3 Byte.
- D. 4 Byte.

4.8. Đối với kiểu dữ liệu Word bộ nhớ lưu trữ một giá trị là:

- A. 8 Byte.
- B. 6 Byte.
- C. 4 Byte.
- D. 2 Byte.

4.9. Đối với kiểu dữ liệu longint bộ nhớ lưu trữ một giá trị là:

- A. 8 Byte.
- B. 6 Byte.
- C. 4 Byte.
- D. 2 Byte.

4.10. Kiểu dữ liệu Real có phạm vi giá trị là:

- A. 0 đến 10^{38}
- B. Giá trị tuyệt đối trong phạm vi từ 10^{-38} đến 10^{39}
- C. Giá trị tuyệt đối trong phạm vi từ 10^{-38} đến $10^{39}-1$
- D. 0 hoặc giá trị tuyệt đối trong phạm vi từ 10^{-38} đến 10^{38}

Phần nâng cao.

4.11. Kiểu dữ liệu Extended có phạm vi giá trị là:

- A. 0 đến 10^{39}
- B. Giá trị tuyệt đối trong phạm vi từ 10^{-38} đến 10^{39} .
- C. 0 hoặc giá trị tuyệt đối trong phạm vi từ 10^{-4932} đến 10^{4932} .
- D. Giá trị tuyệt đối trong phạm vi từ 10^{-4932} đến 10^{4932}

4.12. Đối với kiểu dữ liệu Real bộ nhớ lưu trữ một giá trị là:

- A. 8 Byte.
- B. 6 Byte.
- C. 4 Byte.
- D. 2 Byte.

4.13. Đối với kiểu dữ liệu Extended bộ nhớ lưu trữ một giá trị là:

- A. 10 Byte.
- B. 8 Byte.
- C. 6 Byte.
- D. 4 Byte.

4.14. Kiểu dữ liệu Char có phạm vi giá trị là:

- A. 127 kí tự.
- B. 255 kí tự.
- C. 256 kí tự.
- D. 1024 kí tự.

4.15. Đối với kiểu dữ liệu Char bộ nhớ lưu trữ một giá trị là:

- A. 1 Byte.
- B. 2 Byte.
- C. 3 Byte.
- D. 4 Byte.

4.16. Trong bảng mã ASCII kí tự 'A' có giá trị là:

- A. 32.
- B. 45.
- C. 55.
- D. 65.

4.17. Trong bảng mã ASCII kí tự 'a' có giá trị là:

- A. 80.
- B. 97.
- C. 107.
- D. 117.

4.18. Trong bảng mã ASCII phím Space bar có giá trị là:

- A. 22.
- B. 32.
- C. 42.
- D. 52.

4.19. Kiểu dữ liệu Logic có giá trị là:

- A. 1 kí tự.
- B. True
- C. False
- D. Đáp án B và C.

4.20. Đối với kiểu dữ liệu Logic bộ nhớ lưu trữ một giá trị là:

A. 1 Byte.

C. 3 Byte.

B. 2 Byte.

D. 4 Byte.

4.21. Trong ngôn ngữ lập trình Pascal, kiểu dữ liệu nào trong các kiểu dữ liệu sau có miền giá trị lớn nhất?

A. Byte.

C. Integer.

B. Word.

D. Longint.

II. Phần tự luận.

Hãy chọn phương án trả lời đúng nhất.

Phần cơ bản.

4.1 Trong Turbo Pascal có những kiểu dữ liệu chuẩn nào?

4.2. Trong ngôn ngữ Pascal, có những kiểu nguyên nào thường dùng, phạm vi biểu diễn của mỗi loại?

4.3. Trong ngôn ngữ Pascal, có những kiểu số thực nào thường dùng, phạm vi biểu diễn của mỗi loại?

4.4. Trong ngôn ngữ Pascal, có bao nhiêu kiểu kí tự? Trong ngôn ngữ Pascal, có bao nhiêu kiểu logic, gồm các giá trị nào?

4.5. Vì sao phạm vi biểu diễn của các loại kiểu nguyên khác nhau?

Phần nâng cao.

4.6. Miền giá trị của các loại kiểu thực, số chữ số có nghĩa?

4.7: Muốn tính toán trên các giá trị: 4, 6, 7.5 ta phải sử dụng kiểu dữ liệu gì?

4.8. Hãy cho biết trên kiểu logic có các phép toán nào?

4.9. Hãy cho biết trên kiểu số nguyên có các phép toán nào?

4.10. Hãy cho biết trên kiểu số thực có các phép toán nào?

4.11. Hãy cho biết trên kiểu kí tự có các phép toán nào?

Bài 5

KHAI BÁO BIẾN

A. PHƯƠNG PHÁP GIẢI TOÁN

Cấu trúc chung của khai báo biến:

Trong ngôn ngữ lập trình Pascal, cấu trúc chung của khai báo biến là:

Var

<Danh sách biến 1> : < Kiểu dữ liệu 1>;

<Danh sách biến 2> : < Kiểu dữ liệu 2>;

.....

<Danh sách biến n> : < Kiểu dữ liệu n>;

Trong đó :

- *Danh sách biến* là một hoặc nhiều tên biến. Nếu có nhiều tên biến thì các tên biến được phân cách nhau bởi dấu phẩy.
- *Kiểu dữ liệu* là một trong các kiểu dữ liệu chuẩn của TP.
- Sau từ khoá **VAR** có thể khai báo nhiều danh sách biến khác nhau, tức là cấu trúc :

<Danh sách biến> : <Kiểu dữ liệu>;

có thể xuất hiện nhiều lần.

Chú ý:

- Tất cả các biến đều phải được khai báo.
- Mỗi biến chỉ được khai báo một lần.
- Khi sử dụng biến cần chú ý đặc điểm của các loại dữ liệu gán cho biến.

Ví dụ 1

Giả thiết trong chương trình có các biến thực A,B,C,D,X1,X2 và các biến nguyên N, M, có thể nhận giá trị trong phạm vi từ 20000 đến +25000. Khi đó khai báo biến có thể có dạng:

```
VAR    A, B, C, X1, X2, D : REAL;  
        M, N : INTEGER;
```

B. BÀI TẬP ÁP DỤNG

I. Bài tập trắc nghiệm

Hãy chọn phương án trả lời đúng nhất.

Phần cơ bản

5.1. Để sử dụng biến ta phải:

A. Đặt tên biến.

C. Khai kiểu dữ liệu của biến.

B. Khai báo biến.

D. Tất cả các công việc trên.

5.2. Cú pháp để khai báo biến là:

A. Var <danh sách biến>;

B. <Danh sách biến>: Kiểu dữ liệu;

C. Var <Danh sách biến>: <Kiểu dữ liệu>;

D. Var <Danh sách biến> = <Kiểu dữ liệu>;

5.3. Trong khai báo biến, nếu có nhiều biến có cùng một kiểu thì giữa các biến cách nhau bởi:

A. Dấu chấm (.)

C. Dấu phẩy (,)

B. Dấu chấm phẩy (;)

D. Dấu hai chấm (:)

5.4. Trong khai báo biến, nếu có nhiều biến có các kiểu khác nhau thì giữa các biến cách nhau bởi:

A. Dấu chấm (.)

C. Dấu phẩy (,)

B. Dấu chấm phẩy (;)

D. Dấu hai chấm (:)

5.5. Để khai báo các biến A, B có kiểu kí tự, biến C, D có kiểu số thực, biến E có kiểu số nguyên ta chọn kết quả:

A. Var A, B: Char; E: Word; C, D: Byte.

B. Var A, B: Char; E: Word; C, D: Real;

C. A, B: Char; E: Word; C, D: Real;

D. Var A, B: Char; E: Word; C, D: Longint;

Phần nâng cao.

5.6. Trong khai báo biến:

Var Ch: Char;

A, B: Integer;

C, D: Byte;

E, F: Real;

Tổng bộ nhớ dành cho các biến đã khai báo là:

A. 19.

B. 21.

C. 25.

D. 29.

5.7. Trong các khai báo sau khai báo nào đúng?

A. Khai báo biến m có kiểu thực: Var m: read;

B. Khai báo hai biến p1, p2 có kiểu byte: Var p1:byte, p2:byte;

C. Khai báo biến a và b có kiểu word: Var a: word; b: word;

D. Khai báo biến a và b có kiểu word: Var a: word; b: Byte;

5.8. Trong các khai báo sau, khai báo nào sai?

A. Var a; b: Byte;

C. Var p, q: Extended;

B. Var m, n: char;

D. Var f, k: Boolean;

5.9. Trong khai báo biến:

Var Ch: Char;
 A, B: Integer;
 C, D: Byte;
 E, F: Boolean;

Các giá trị mà biến nhận được dưới đây giá trị nào đúng?

- A. Ch = 'A'; a = 5; b = 6; c = 8; d = 10; e = 1; f = 2.
- B. Ch = 'B'; a = 0; b = 3; c = 6; d = 10; e = 1; f = True.
- C. Ch = 'O'; a = 0; b = 3; c = 6; d = 10; e = 1; f = False.
- D. Ch = '1'; a = 2; b = 4; c = 6; d = 10; e = True; f = False.

5.10. Trong các phát biểu dưới đây, phát biểu nào đúng?

- A. Khai báo biến a có kiểu byte thì biến a chiếm 2 byte trong bộ nhớ.
- B. Khai báo biến M có kiểu Longint thì biến M chiếm 4 byte trong bộ nhớ.
- C. Khai báo biến a có kiểu boolean thì biến a không chiếm byte nào trong bộ nhớ.
- D. Khai báo biến h có kiểu word thì biến h chiếm 4 byte trong bộ nhớ.

II. Phần tự luận

5.1. Nếu ch là một biến kiểu char thì các hàm sau đây chr(ord(ch)), ord(chr(ch)) cho kết quả gì? Tại sao?

5.2. Tìm công thức để đổi một kí tự hoa thành kí tự thường và ngược lại.

5.3. Nếu khai báo m, n, k là các biến kiểu byte, biết rằng m=100, n=200.

Hãy cho biết giá trị của k trong các trường hợp sau:

- a. $k = m \text{ div } 2 + n$. (div là phép chia nguyên).
- b. $k = 3 * m + 2 * n$.
- c. $k = 1.5 * m + n$.

5.4. Hãy so sánh các kí tự sau:

- a. Kí tự 'A' và 'a'.
- b. Kí tự 'A' và '1'.
- c. Kí tự 'A' và kí tự trắng.

5.5. Tính tổng bộ nhớ dành cho các biến đã khai báo:

Var Ch: Char;
 A, B: Integer;
 C, D: Byte;
 E, F: Real;
 P, K: Boolean;
 L, M: Extended;

Bài 6

PHÉP TOÁN, BIỂU THỨC, CÂU LỆNH GÁN

A. PHƯƠNG PHÁP GIẢI TOÁN

1. Các phép toán số học

Xác định được các phép toán trên các kiểu dữ liệu khác nhau.

- Các phép toán số học với số nguyên: +, -, *, DIV, MOD.

- Các phép toán số học với số thực: +, -, *, /.

- Các phép toán quan hệ: <, <=, >, >=, =, <>. Dùng để so sánh hai đại lượng, kết quả của các phép toán này là true hoặc false.

- Các phép toán Logic: NOT, OR, AND, XOR thường dùng để tạo các biểu thức logic từ các biểu thức quan hệ đơn giản.

Bảng các phép toán thường dùng:

Tên phép toán	Kí hiệu toán học	Kí hiệu trong TP	Loại phép toán
Cộng	+	+	Số học
Trừ	-	--	
Nhân	X	*	
Chia	/	/	
Chia nguyên		DIV	
Lấy phần dư		MOD	
Nhỏ hơn	<	<	Quan hệ
Nhỏ hơn hoặc bằng	≤	<=	
Lớn hơn	>	>	
Lớn hơn hoặc bằng	≥	>=	
Bằng	=	=	
Khác	≠	<>	
Phủ định	¬	NOT	Logic
Hoặc (Tuyển)	∨	OR	
Và (Hội)	∧	AND	

2. Biểu thức số học

- Biểu thức số học là biểu thức nhận được từ các hằng số, biến số và hàm số liên kết với nhau bằng các phép toán số học.

- Thứ tự thực hiện biểu thức số học: Trong ngoặc trước, ngoài ngoặc sau. Trong dãy các phép toán không chứa ngoặc thì thực hiện từ trái sang phải theo thứ tự của các phép toán: nhân, chia, chia lấy nguyên, chia lấy dư thực hiện trước và các phép toán cộng, trừ thực hiện sau.

- Hàm số học chuẩn thông dụng.

Hàm	Kiểu đối số	Kiểu hàm số
Bình phương: SQR(X)	Nguyên hoặc thực	Kiểu của đối số
Căn bậc hai: SQRT(X)	Nguyên hoặc thực	Thực
Giá trị tuyệt đối: ABS(X)	Nguyên hoặc thực	Kiểu của đối số
Sin(X)	Nguyên hoặc thực	Thực
Cos(X)	Nguyên hoặc thực	Thực
Logarit tự nhiên Ln x ln(x)	Nguyên hoặc thực	Thực
Lũy thừa của số e e ^x exp(x)	Nguyên hoặc thực	Thực

- Hai biểu thức có cùng kiểu dữ liệu được liên kết với nhau bởi phép toán quan hệ cho ta một biểu thức quan hệ.

<biểu_thức_1> <phép_toán_quan_hệ> <biểu_thức_2>

- Thứ tự thực hiện:

+ Tính giá trị các biểu thức.

+ Thực hiện phép toán quan hệ.

+ Kết quả của biểu thức quan hệ là giá trị logic True (*Đúng*) hoặc False (*Sai*).

- Các biểu thức quan hệ liên kết với nhau bởi phép toán logic ta được biểu thức logic. Biểu thức logic nhận giá trị True hoặc False.

Các phép toán **AND** và **OR** dùng để kết hợp nhiều biểu thức logic hoặc quan hệ, thành một biểu thức diễn tả các điều kiện phức tạp.

Giả thiết, các biến **A** và **B** là những biến logic. Để thuận tiện, ta sẽ kí hiệu giá trị logic True là 1 và False là 0. Khi đó, giá trị các biểu thức logic:

NOT A, **A AND B** và **A OR B** được xác định trong các bảng sau :

+ Bảng giá trị phép tính logic:

A	0	1
NOT A	1	0

A	0	0	1	1
B	0	1	0	1
A AND B	0	0	0	1
A OR B	0	1	1	1

3. Lệnh gán

Lệnh gán là một trong những lệnh cơ bản nhất của một ngôn ngữ lập trình bất kì nói chung và ngôn ngữ Turbo Pascal (TP) nói riêng.

Trong TP câu lệnh gán có dạng:

<Tên biến> := <Biểu thức>;

trong trường hợp đơn giản, *Tên biến* là tên của biến đơn. Kiểu của giá trị biểu thức phải phù hợp với kiểu của biến.

Chức năng của lệnh gán là tính giá trị của biểu thức và ghi giá trị đó vào địa chỉ của biến có tên ở vế trái dấu $:=$ tức là gán giá trị cho biến.

Lệnh gán được thực hiện qua các bước sau :

- Tính giá trị của biểu thức ở vế phải.
- Gán giá trị nhận được cho biến ở vế trái.

Ví dụ 1

$X1 := (-B - \text{SQRT}(B*B - 4*A*C)) / (2*A) ;$

$X2 := -B/A - X1 ;$

Chú ý:

- Dấu chấm phẩy “;” dùng để phân biệt một câu lệnh với câu lệnh tiếp theo.

B. BÀI TẬP ÁP DỤNG

I. Bài tập trắc nghiệm

Hãy chọn phương án trả lời đúng nhất.

Phần cơ bản

6.1. Các biểu diễn của phép toán số học với số nguyên là:

- A. Cộng (+) và trừ (-).
- B. Nhân (*) và chia (/).
- C. Cộng (+), trừ (-), nhân (*), chia lấy nguyên (div) và chia lấy dư (mod).
- D. Cộng (+), trừ (-), nhân (*), chia (/).

6.2. Các biểu diễn của phép toán số học với số thực là:

- A. Cộng (+) và trừ (-).
- B. Nhân (*) và chia (/).
- C. Cộng (+), trừ (-), nhân (*), chia (:).
- D. Cộng (+), trừ (-), nhân (*), chia (/).

6.3. Các biểu diễn của phép toán quan hệ là:

- A. Bằng (=), lớn hơn (>), bé hơn (<).
- B. Khác (<>).
- C. Lớn hơn hoặc bằng (>=), bé hơn hoặc bằng (<=).
- D. Tất cả các phép toán trên.

6.4. Các biểu diễn của phép toán logic là:

- A. Not, xor, or.
- B. And, or, not, xor.
- C. And, or, xor.
- D. And, or.

6.5. Trong ngôn ngữ lập trình Pascal, thứ tự của các phép toán được thực hiện trong biểu thức số học là:

- A. Thực hiện các phép toán trong ngoặc trước, ngoài ngoặc sau.
- B. Thực hiện các phép toán trong ngoặc trước, ngoài ngoặc sau, trong dãy các phép toán không chứa ngoặc thì thực hiện từ trái sang phải, theo thứ tự các phép toán nhân, chia, chia nguyên, lấy phần dư thực hiện trước và các phép toán cộng trừ thực hiện sau.
- C. Thực hiện các phép toán trong ngoặc trước, ngoài ngoặc sau, nhân chia trước, cộng trừ sau.
- D. Thực hiện các phép toán trong ngoặc trước, ngoài ngoặc sau, nhân chia trước, cộng trừ sau, chia nguyên, lấy phần dư thực hiện cuối cùng. Thứ tự thực hiện từ trái sang phải.

Phần nâng cao.

6.6. Trong Pascal các biểu diễn nào dưới đây là đúng?

- A. $5a - 4c$; $(12*a + 5b)/2$; $(x-y)*z$;
- B. $\text{delta} = b^2 - 4ac$; $x_1 = x_2 = -b/2a$;
- C. $1*a*(1-a)+2$; $13*a - (1/x) + (x-y)*(y-x)$;
- D. $(a-b)/(c-d) + (a-1/c)$; $2x^2 - 4x + 2$;

6.7. Trong Pascal biểu diễn hàm nào dưới đây đúng?

- A. $\text{Sqr}(x)$; $\text{Sqrt}(x)$; $\sin x$;
- B. $\text{Sqrt}(x)$; $\text{abs}(x)$; $\ln(x)$;
- C. $\text{Sqr}(x)$; $\text{Sqrt}(x)$; $\text{abs}(x)$;
- D. $\text{Cos}(x)$; $e^x(x)$; $\log(x)$;

6.8. Trong Turbo Pascal các biểu diễn dưới đây biểu diễn nào đúng?

- A. $(a-b) > (c-d)$; $(a-b) <> (b-a)$; $12a > 5a$;
- B. $\text{sqrt}(a-b) > x$; $(1/x - y) >= 2*x$; $3*a > 3$;
- C. $b^2 > ac$; $a(1-a) + (a-b) >= 0$; $1/x - x < 0$;
- D. $(a-b) > (c-d)$; $(1/x - y) >= 2*x$; $bb > ac$;

6.9. Cho các khai báo:

Var Ch: Char;
A, B: Integer;
C, D: Byte;
E, F: Real;
P, K: Boolean;
L, M: Extended;

Các phép gán nào dưới đây là đúng?

- A. $\text{Ch} := \text{"abc"};$ $a := 2;$ $b := 5;$ $F := 3.14;$

- | | | |
|--------------|-----------------|----------|
| B. ch:= 'a'; | a:= 2; b:=4; | c:=6; |
| C. p:=true; | ch:=a; a:=1000; | f:=3,14; |
| D. k:=false; | ch = a; b:=50; | e:=120; |

6.10. Cho các khai báo:

```
Var   Ch: Char;
      A, B: Integer;
      C, D: Byte;
      E, F: Real;
```

Các phép gán nào dưới đây là đúng?

- | | | |
|-----------|--------|----------|
| A. A:=b; | b:=c; | d:=f; |
| B. c:=d; | e:=f; | ch:=c; |
| C. A:=b; | c:=d; | e:=f; |
| D. A:= 2; | b:= a; | e:=3.14; |

6.11. Trong ngôn ngữ lập trình Pascal, về mặt cú pháp câu lệnh nào sau đây là sai?

- | | |
|---------------------|--------------|
| A. x:=12,5; | C. c:=PI*12; |
| B. b:=(a=5)or(c=7); | D. a:=3.12 |

6.12. Trong ngôn ngữ lập trình Pascal, hằng xâu kí tự "Hoa co mua xuan" được viết như thế nào?

- | | |
|----------------------|----------------------|
| A. 'Hoa co mua xuan' | C. "Hoa co mua xuan" |
| B. Hoa co mua xuan | D. 'Hoa co mua xuan' |

6.13. Trong ngôn ngữ lập trình Pascal, với câu lệnh như sau:

Writeln("KQ la:",a); sẽ ghi ra màn hình kết quả gì?

- | | |
|-----------------|---------------------------|
| A. Ket qua la a | C. KQ la: <giá trị của a> |
| B. KQ la a | D. Câu lệnh sai. |

6.14. Trong ngôn ngữ lập trình Pascal, với câu lệnh như sau:

Writeln('KQ la:',a); sẽ ghi ra màn hình kết quả?

- | | |
|-----------------|---------------------------|
| A. Ket qua la a | C. KQ la: <giá trị của a> |
| B. KQ la a | D. Không đưa ra gì cả. |

6.15. Trong ngôn ngữ lập trình Pascal, với câu lệnh như sau (a là một biến kiểu số thực):

a:=12;

Writeln("KQ la:", a);

sẽ ghi ra màn hình kết quả?

- | | |
|-------------------|-----------------|
| A. KQ là: 1.2E+01 | C. KQ là 12 |
| B. KQ la: a | D. Câu lệnh sai |

II. Phần tự luận.

6.1. Khai báo các biến và thực hiện biểu diễn cho cách tính delta, nghiệm trong các trường hợp của phương trình bậc hai $ax^2 + bx + c = 0$.

- 6.2. Khai báo biến và biểu diễn giá trị của nghiệm x trong phương trình bậc nhất $ax+b=0$.
- 6.3. Khai báo 2 biến kiểu nguyên, 3 biến kiểu thực, 1 biến kiểu boolean và gán giá trị cho chúng.
- 6.4. Khi thực hiện phép gán giá trị cho biến cần chú ý điều gì?
- 6.5. Khi thực hiện các phép toán trên các kiểu dữ liệu chuẩn cần chú ý điều gì?

Bài 7

CÁC THỦ TỤC CHUẨN VÀO / RA ĐƠN GIẢN

A. PHƯƠNG PHÁP GIẢI TOÁN

Lập chương trình bằng ngôn ngữ lập trình và khi thực hiện thì người sử dụng luôn truyền giá trị vào cho chương trình thông qua các biến. Quá trình đó gọi là nhập dữ liệu. Khi xử lý xong sẽ hiển thị kết quả ra màn hình, quá trình đó gọi là xuất dữ liệu. Turbo Pascal cung cấp các thủ tục chuẩn để nhập và xuất dữ liệu.

Các câu lệnh nhập vào và đưa ra được gọi chung là các *lệnh vào/ ra*. Các *lệnh vào/ ra* là một bộ phận không thể thiếu của một chương trình Pascal hoàn chỉnh.

Trong phần này, ta sẽ xét các *lệnh vào/ra đơn giản* phục vụ việc nhập thông tin vào từ bàn phím và đưa thông tin ra màn hình.

1. Nhập dữ liệu:

- Dùng để nhập nhiều bộ dữ liệu khác nhau vào cho một chương trình xử lý.

- Cú pháp nhập: Read/Readln(<tên_biến_1>, ..., <tên_biến_k>);

2. Xuất dữ liệu

- Dùng để đưa kết quả ra màn hình sau khi xử lý để người sử dụng thấy.

- Cú pháp xuất: Write/Writeln(<tham_số_1>, ..., <tham_số_k>);

Sự khác nhau cơ bản giữa Read với Readln và Write với Writeln là con trỏ màn hình xuất nhập xuống dòng hay không xuống dòng.

Chú ý

- Các lệnh **READLN**, **WRITELN** có thể không có tham số.
- Trong lệnh **WRITE** hoặc **WRITELN** sau mỗi kết quả ra (Biến, hằng, biểu thức) có thể có *quy cách ra*. Quy cách ra có dạng :
 - Đối với kết quả thực : <Độ rộng> : <Số chữ số phần phân>
 - Đối với các kết quả khác: <Độ rộng>

trong đó *Độ rộng* và *Số chữ số phần phân* là các hằng nguyên.

Ví dụ:

```
WRITELN(N:5, X:6:2);
```

```
WRITE(I:3,J:4, A+B:8:3);
```

Ở câu lệnh thứ nhất, trên màn hình 5 vị trí kể từ vị trí con trỏ màn hình sẽ được dành để đưa ra giá trị N, nếu N có giá trị nguyên ít hơn 5 chữ số hoặc giá trị âm ít hơn 4 chữ số thì các vị trí đầu tiên sẽ được điền dấu cách. Tiếp theo đó là 6 vị trí dành để đưa ra X, trong đó 2 vị trí dành để đưa ra phần thập phân, các phần nguyên và thập phân tách nhau bởi dấu chấm. Như vậy phần nguyên còn lại 3 vị trí. Các vị trí ở đầu, nếu còn thừa sẽ được điền dấu cách. Ở câu lệnh thứ 2 đưa ra I trên 3 vị trí, J trên 4 vị trí và kết quả A+B trên 8 vị trí, trong đó có 3 vị trí dành cho phần thập phân.

B. BÀI TẬP ÁP DỤNG

I. Bài tập trắc nghiệm

Hãy chọn phương án trả lời đúng nhất.

Phần cơ bản

7.1. Các thủ tục chuẩn vào/ra dùng để làm gì?

- A. Dùng để gán giá trị cho biến.
- B. Dùng để đưa dữ liệu vào và xuất dữ liệu ra cho biến.
- C. Dùng để nhập dữ liệu vào từ bàn phím và xuất dữ liệu ra ngoài.
- D. Dùng để xuất nhập câu lệnh.

7.2. Để nhập dữ liệu ta sử dụng thủ tục:

- A. Read(danh sách biến);
- B. Readln(danh sách biến);
- C. Real(danh sách biến);
- D. Đáp án A và B đều đúng.

7.3. Để đưa dữ liệu ra màn hình ta sử dụng thủ tục:

- A. Write(Danh sách giá trị);
- B. Wite(Danh sách giá trị);
- C. Writeln(Danh sách giá trị);
- D. Đáp án A và C đều đúng.

7.4. Khi nào sử dụng thủ tục readln?

- A. Khi cần xuất dữ liệu vào biến.
- B. Khi cần lấy giá trị của một biến.
- C. Khi cần để nhập dữ liệu, sau khi nhập xong thì con trỏ xuống dòng và về đầu dòng.
- D. Khi có thủ tục Writeln.

7.5. Thủ tục Write(bien) và Writeln(bien) khác nhau cơ bản là:

- A. Không có sự khác nhau.

B. Thủ tục Write sau khi viết giá trị ra thì con trỏ không chuyển xuống đầu dòng tiếp theo, ngược lại thủ tục writeln thì viết xong và con trỏ xuống dòng tiếp theo.

C. Thủ tục Write dừng khi viết nhiều giá trị còn writeln thì dừng để viết một giá trị.

D. Thủ tục write đi kèm với thủ tục read còn thủ tục writeln đi kèm với thủ tục readln.

7.6. Cú pháp để nhập dữ liệu vào từ bàn phím là:

A. Read(<danh sách biến vào>; C. Readln[<danh sách biến>];

B. Readln(<danh sách biến vào>; D. Đáp án A và B đều đúng.

7.7. Cú pháp để xuất dữ liệu ra màn hình là:

A. Write[<danh sách giá trị>; C. Writeln(<danh sách giá trị>;

B. Write(<danh sách giá trị>; D. Đáp án B và C đều đúng.

Phần nâng cao

7.8. Trong cú pháp Read(<danh sách biến vào>) hay Writeln(<danh sách giá trị>) thì giữa các biến hay các giá trị cách nhau bởi:

A. Dấu hai chấm (:); C. Dấu phẩy (,)

B. Dấu chấm phẩy (;) D. Dấu chấm (.)

7.9. Kết quả của đoạn chương trình sau là gì?

```
Var    i:byte;
```

```
Begin
```

```
    I:= 12345;
```

```
    Writeln(I);
```

```
    Write(-456);
```

```
End.
```

A. 12345 -456

B. 12345

 -456

C. -456 12345

D. -456

 12345

7.10. Kết quả của đoạn chương trình sau là gì?

```
Var    i:byte;
```

```
Begin
```

```
    Writeln('Ket qua la:');
```

```
    I:= 123;
```

```
Write(1:2);
Write(-456);
```

End.

- A. Ket qua la:
123-456
- B. Ket qua la: 123 -456
- C. 123 -456.
- D. 123
-456

7.11. Kết quả của đoạn chương trình sau là gì?

```
Var a,b: byte;
```

```
Begin
```

```
a:=50; b:= 25;
```

```
Write(a,b);
```

```
Writeln;
```

```
Writeln(a+b);
```

```
Write('chuc thanh cong');
```

End.

- A. 50 25 75 chuc thanh cong
- B. 50 25 75
chuc thanh cong
- C. 50 25
75
chuc thanh cong
- D. 50 25
75 chuc thanh cong

7.12. Cho đoạn chương trình:

```
Var a,b: real;
```

```
Begin
```

```
a:=12345.6789; b:=12.12345678;
```

```
Write(a,b:3:4);
```

```
Writeln;
```

```
Write('chuc thanh cong');
```

End.

Kết quả của chương trình là:

- A. 12345.6789 12.12345678 chuc thanh cong
- B. 12345.6789 12.12345678

chuc thanh cong

C. 12345.6789

12.12345678

chuc thanh cong

D. 12345.6789 12.1234

chuc thanh cong

7.13. Cho đoạn chương trình:

Var ch: Char;

Begin

Ch:= 'I'; Write(ch);

Writeln('LOVE');

Write('YOU');

Writeln;

End.

Kết quả khi chạy chương trình là

A. I LOVE YOU

B. I LOVE

YOU

C. I

LOVE

YOU

D. I

LOVE YOU

7.14. Cho đoạn chương trình

Var ok: Boolean;

Begin

Ok:= 3 > 5;

Write(ok);

Write(5>3);

End.

Kết quả khi chạy chương trình là:

A. ok 5>3

B. FalseTrue

C. okTrue

D. False ok

7.15. Cho đoạn chương trình:

```
Var a,b: Byte;  
Begin  
    a:=2; b:=6;  
    Write('biểu thức a>b là ',a>b);  
End.
```

Kết quả khi chạy chương trình là:

- A. a>b
- B. biểu thức là a>b
- C. false
- D. biểu thức a>b là false

II. Phần tự luận

Phần cơ bản

- 7.1. Có thể dùng thủ tục Readln để đọc một hàng được không? Vì sao?
- 7.2. Có thể sử dụng readln có quy cách không? Vì sao? Ví dụ Readln(ii:4);
- 7.3. Cho biết ý nghĩa của việc viết các giá trị theo quy cách.
- 7.4. Viết chương trình nhập vào hai số nguyên, tính tổng của hai số nguyên đó và hiển thị kết quả ra màn hình.
- 7.5. Hãy viết ra màn hình một kí tự, sau đó viết ra mã số ASCII của nó theo dạng sau: 'A': 65.
- 7.6. Lập chương trình tính diện tích S của hình thang với đáy dài a, đáy ngắn b, chiều cao h nhập từ bàn phím.

Phần nâng cao

- 7.7. Cho biết kết quả và kiểu dữ liệu của các biểu thức sau:
 - a. $3 + 5.0$
 - b. $6/3 + 2 \text{ div } 3$
 - c. $(10*((45 \bmod 3) + 12))/6$
 - d. $(5 \leq 3) \text{ and } (\text{not true and } (12 \text{ div } 3 \leq 1))$
- 7.8. Một người có số tiền là A đồng, lãi suất tiết kiệm là s% một tháng. Hỏi sau t tháng người đó được cả vốn và lãi là bao nhiêu? Lập chương trình với số tiền A là 20 triệu, s=0.7%, t=30 tháng, các giá trị này nhập từ bàn phím.
- 7.9. Cho tam giác ABC có ba đỉnh có tọa độ là (x1,y1), (x2,y2), (x3,y3). Viết chương trình tính diện tích và chu vi của tam giác đó.
- 7.10. Cho a,b,c là độ dài ba cạnh của một tam giác được nhập từ bàn phím. Tính chu vi, diện tích, độ dài ba đường cao, ba đường phân giác của tam giác..
- 7.11. Viết chương trình tính diện tích mặt cầu và thể tích hình cầu. Cho biết bán kính R nhập vào từ bàn phím.
- 7.12. Tính chu vi của đa giác đều n cạnh nội tiếp đường tròn bán kính R.
- 7.13. Viết chương trình tính tổng của một cấp số cộng có n phần tử biết phần tử thứ nhất là A và công sai là D.

Bài 8

SOẠN THẢO, DỊCH, THỰC HIỆN VÀ HIỆU CHỈNH CHƯƠNG TRÌNH

A. PHƯƠNG PHÁP GIẢI TOÁN

1. Khởi động Turbo.

- Chuyển vào thư mục chứa file Turbo.exe
- Gõ turbo.exe và nhấn phím enter.
(nếu ở môi trường Win thì chỉ cần bấm biểu tượng Turbo Pascal)

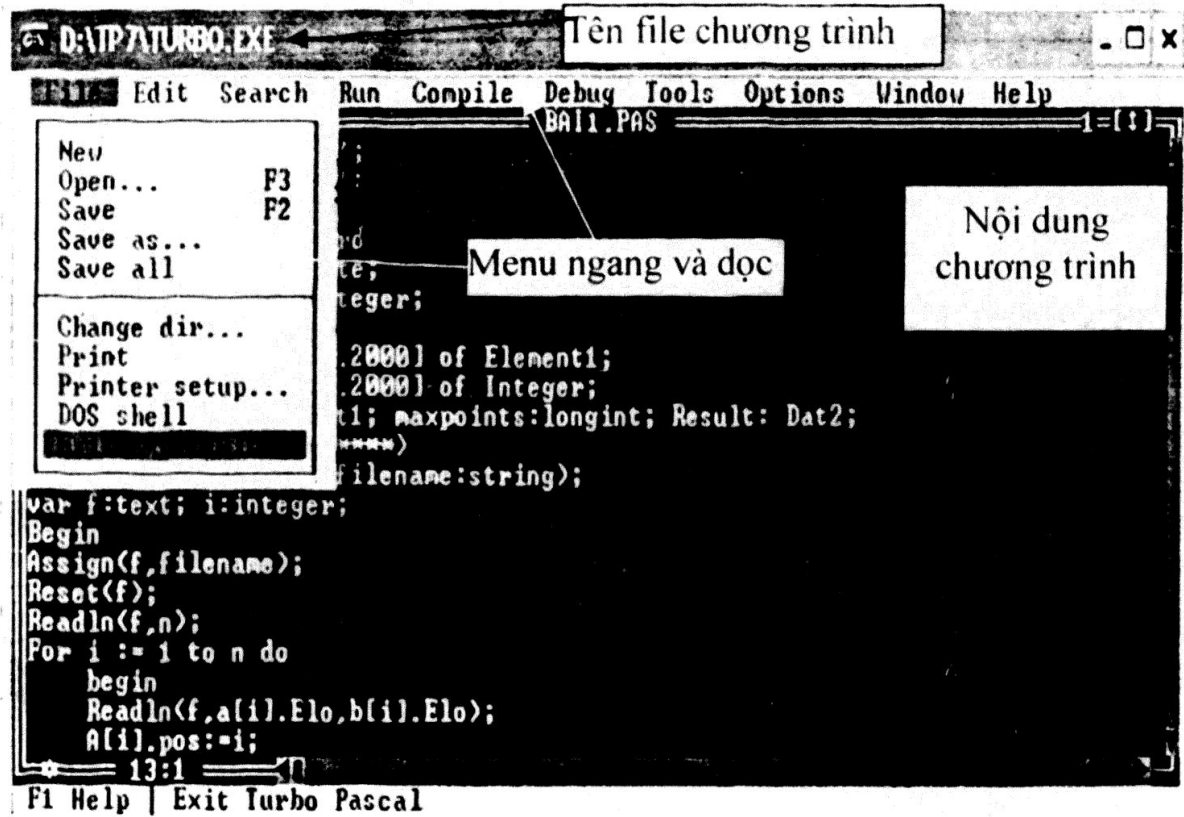
2. Dịch lỗi cú pháp. Thực hiện chương trình.

- Gõ các lệnh của chương trình (giống như trong hệ soạn thảo văn bản).
- Lưu file chương trình lên đĩa bấm F2, đặt tên.
- Biên dịch lỗi cú pháp: bấm ALT_F9. Thông thường khi dịch chương trình bị lỗi chỗ nào thì máy dừng lại chỗ đó với một vệt sáng đỏ, con trỏ màn hình đậu tại vị trí lỗi. Tiến hành sửa lỗi và dịch lại cho đến hết.

3. Tìm lỗi thuật toán và hiệu chỉnh

- Thực hiện chương trình: Bấm CTR_F9
- Xây dựng test.
- Nhập dữ liệu, thu kết quả, đối chứng với kết quả của test.

4. Màn hình soạn thảo của Turbo Pascal.



+ Màn hình soạn thảo gồm:

+ Hệ thống Menu: File, Edit, Search, Run....

+ Thanh cuộn ngang, dọc.

+ Màn hình soạn thảo.

+ Một số phím và tổ hợp phím sử dụng trong soạn thảo:

- F2: lưu; F3: Mở file; Alt + X: thoát khỏi môi trường soạn thảo;

- Sao chép: Ctrl + Ins; Dán: Shift + Ins; Xoá: Ctrl + Del;

- Chọn vùng: Shift và các phím khác; Huỷ chọn: Ctrl + K+H;

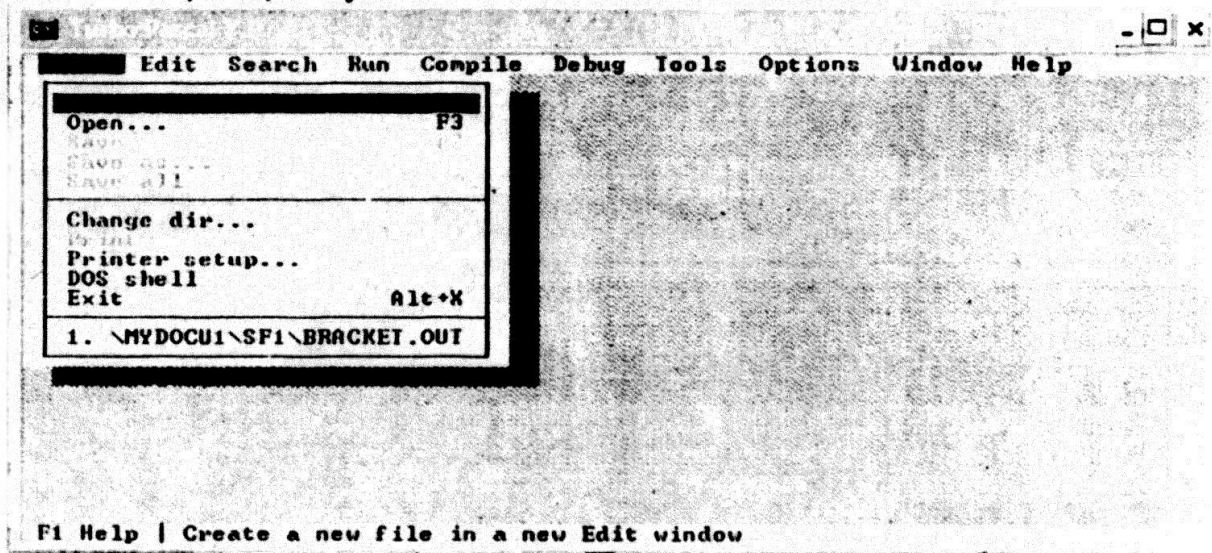
- Xoá một dòng: Ctrl + Y. Dịch từng bước: F8; F5: Chọn cửa sổ soạn thảo hiện thời làm cửa sổ duy nhất; F6: Thay đổi cửa sổ hiện thời; Chạy chương trình: Ctrl + F9.

5. Một số chức năng khác

Bảng chọn (Menu) của Turbo Pascal (TP) được kích hoạt bằng cách nhấn phím *F10*. Khi nhấn phím mũi tên → hay ← ta có thể chuyển tới các mục khác nhau của bảng chọn. Khi nhấn phím *Enter* mục được chọn sẽ được kích hoạt và ta sẽ có một bảng chọn dọc (Menu Pop-Up) xuất hiện, giới thiệu các mục công việc. Các mục thường được dùng là **File** (Tập), **Run** (Thực hiện), **Debug** (Gỡ rối) và **Option** (Tuỳ chọn).

a) Mục File

Khi chọn mục này ta có màn hình như ở hình dưới:



Các công việc thường được chọn ở đây là :

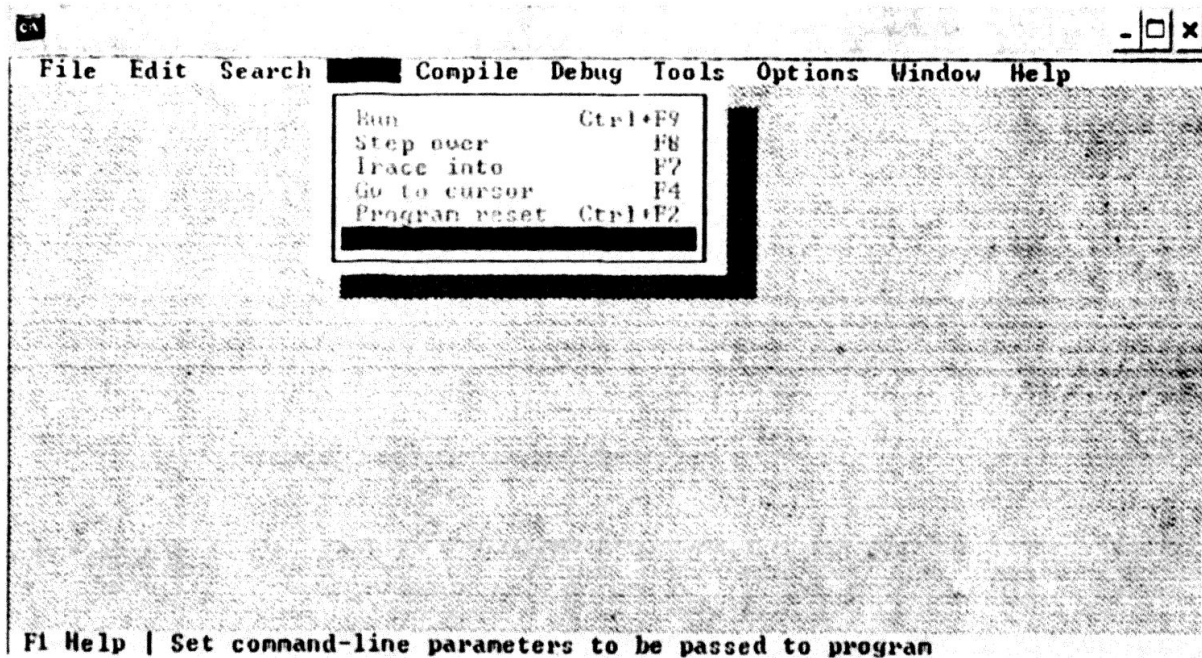
- New - Mở cửa sổ mới để soạn thảo chương trình.
- Open - Mở tệp đã có trên đĩa. Có thể vào mục này bằng cách nhấn phím *F3*. Trên màn hình sẽ xuất hiện cửa sổ để khai báo tên tệp cần mở.

- Save - Để lưu tệp đang soạn thảo. Nếu đây là tệp chưa đặt tên thì TP sẽ hỏi tên tệp để lưu trữ. Có thể vào mục này bằng cách nhấn phím *F2*.
- Save as - Để lưu tệp đang soạn thảo với tên mới.
- Save all - Để lưu tất cả các tệp trên tất cả các cửa sổ đang mở. TP 7.0 cho phép mở nhiều cửa sổ đồng thời.
- Changer dir - Thay đổi thư mục chủ cho TP.
- DOS shell - Chuyển sang màn hình DOS, ở đó có thể thực hiện các lệnh của MS DOS. Để quay trở về màn hình TP cần gõ lệnh EXIT.
- Exit - Thoát khỏi TP. Có thể thoát khỏi TP bằng cách gõ *Alt + X*.

Trong một số hoàn cảnh cụ thể, có thể có những mục chưa được phép chọn. Những mục này sẽ hiển thị dưới dạng mờ hơn các mục được quyền chọn.

b) Mục Run

Khi chọn mục này ta được cửa sổ như ở hình sau đây. Mục này dùng để xác định chế độ thực hiện chương trình.



Các chế độ có thể chọn là:

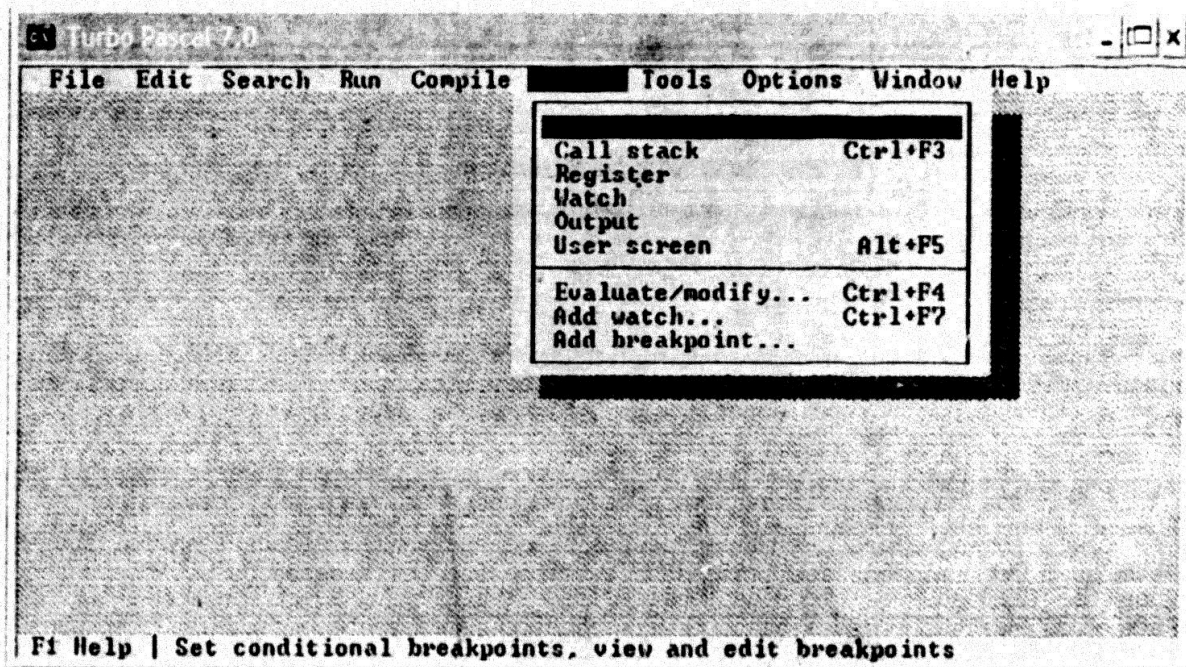
- Run - Thực hiện chương trình nếu trước đó chương trình đã được dịch bằng cách gõ *Alt + F9* và chương trình không có lỗi. Nếu chương trình chưa được dịch, TP sẽ dịch và thực hiện chương trình khi chương trình nguồn không có lỗi cú pháp. Nếu chương trình nguồn có lỗi cú pháp, TP sẽ thông báo lỗi và con trỏ màn hình sẽ

nhấp nháy ở dòng phát hiện lỗi. Có thể vào chế độ này bằng cách gõ *Ctrl + F9*.

- Step over - Thực hiện từng dòng lệnh trên màn hình soạn thảo. Các lời gọi chương trình con được xem như một lệnh. Cần lưu ý là một dòng lệnh trên màn hình soạn thảo có thể chứa *nhiều câu lệnh* của TP. Có thể vào chế độ này bằng cách gõ *F8*.
- Trace into - Thực hiện từng dòng lệnh trên màn hình soạn thảo, kể cả các dòng lệnh ở chương trình con. Có thể vào chế độ này bằng cách gõ *F7*.
- Go to cursor - Thực hiện chương trình cho đến dòng có con trỏ màn hình. Có thể vào chế độ này bằng cách gõ *F4*.
- Program reset - Xóa các trạng thái của TP đối với chương trình đang thực hiện để chuẩn bị dịch và thực hiện lại từ đầu. Có thể vào chế độ này bằng cách gõ *Ctrl + F2*.

c) Mục Debug

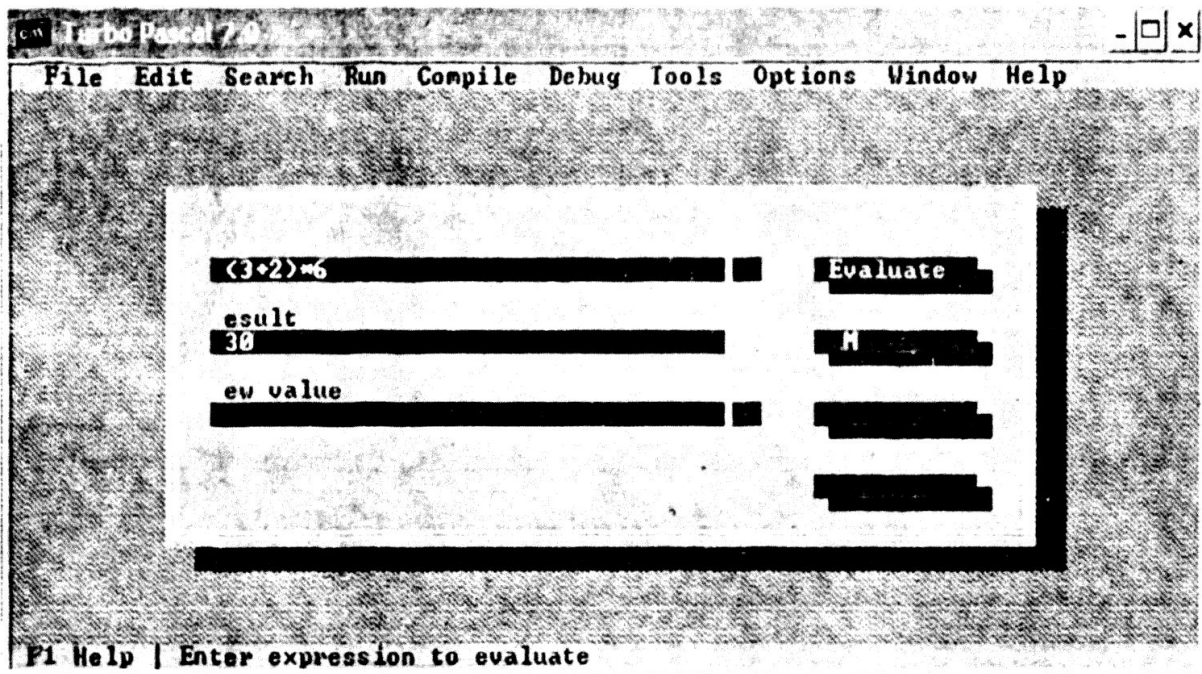
Phục vụ hiệu chỉnh chương trình. Khi chọn mục này ta có cửa sổ như ở hình dưới đây:



Các mục thường sử dụng ở đây là:

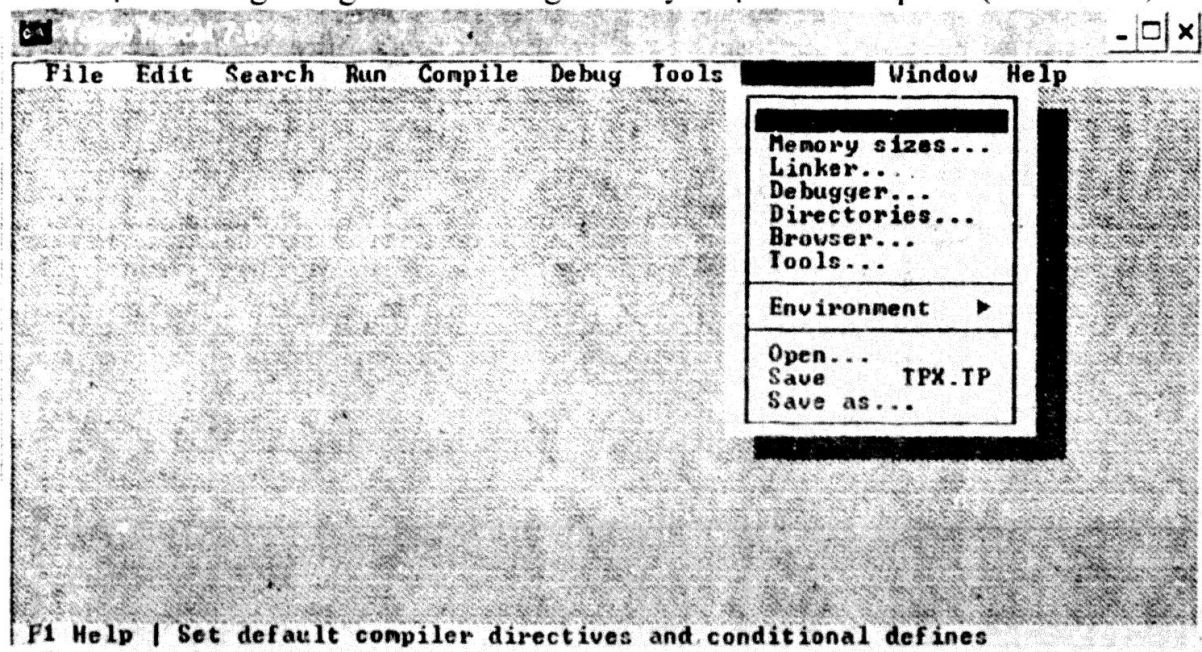
- Evaluate/modify - Dùng để tính giá trị biểu thức. TP sẽ mở cửa sổ để gõ biểu thức cần tính. Sau khi gõ biểu thức, nhấn phím *Enter*. TP sẽ tính và cho giá trị ở cửa sổ giá trị hình dưới.

- Add watch - Dùng để mở cửa sổ theo dõi giá trị biến trong quá trình thực hiện chương trình. Có thể vào mục này bằng cách gõ *Ctrl + F7*.

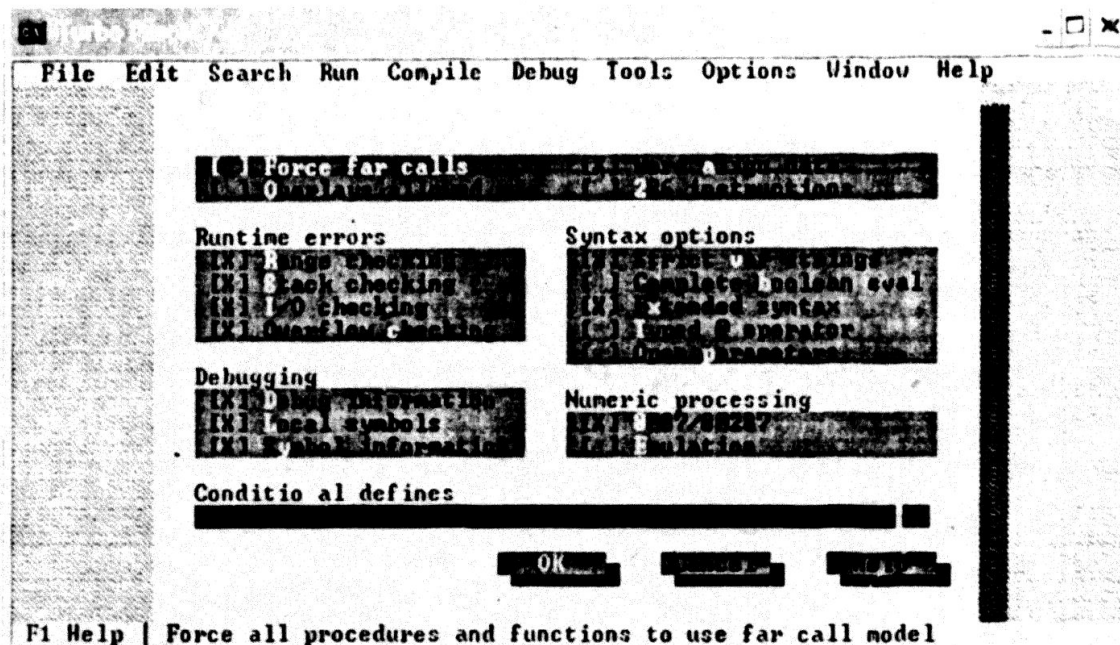


d) Mục Option

Mục thường dùng hơn cả trong các tùy chọn là Compiler (hình dưới):



Khi chọn mục này cửa sổ như trên hình sẽ xuất hiện. Tại đây TP trình ra các chế độ mà ta có thể xác lập cho chương trình dịch. Các chế độ này cũng có thể xác lập ngay trong chương trình nguồn bằng câu lệnh của TP. Các tùy chọn được chia thành nhiều nhóm. Việc chuyển từ nhóm này sang nhóm khác được thực hiện bằng cách gõ phím *Tab*. Việc chọn bỏ một tùy chọn được thực hiện bằng cách gõ *dấu cách*. Những tùy chọn có hiệu lực được đánh dấu **X** (hình dưới).



e) Chuyển cửa sổ màn hình soạn thảo

TP cho phép mở nhiều cửa sổ soạn thảo. Để chuyển sang cửa sổ tiếp theo cần gõ phím *F6*. Để quay lại về cửa sổ trước đó gõ *Shift + F6*.

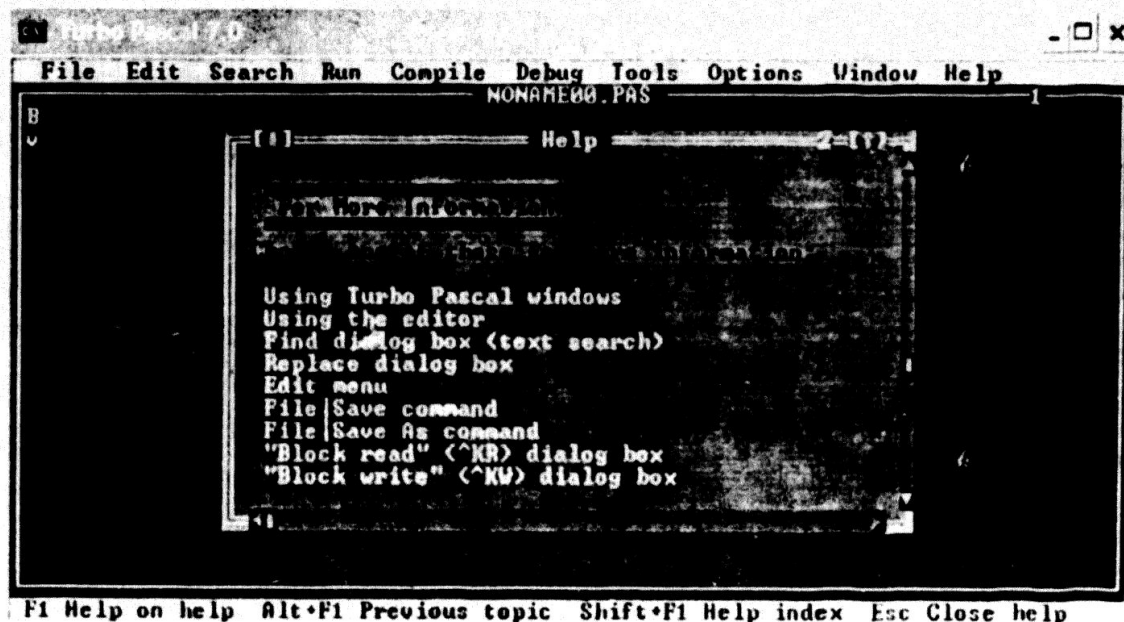
f) Trợ giúp

Để TP có thể đưa ra các giải thích trợ giúp cần có tệp *TURBO.TPH*. Có thể vào trợ giúp bằng cách chọn mục *Help* hoặc gõ phím *F1*. TP sẽ trình ra các mục cần chọn để có thể lựa chọn tiếp vấn đề ta quan tâm.

Nếu cần hiểu sâu một từ khoá hay tên chuẩn, ta có thể để con trỏ màn hình ở vị trí trong hoặc cuối tên và gõ *Ctrl + F1*.

Có thể tra cứu bằng cách gõ *Shift + F1* rồi lần lượt gõ tiếp các kí tự của đối tượng cần tìm hiểu cho đến khi TP tìm được đối tượng đó.

Các phím phục vụ soạn thảo có thể tra cứu bằng cách gõ *F1* vào trợ giúp và cuối nội dung thông tin trợ giúp này sẽ có danh sách các mục có thể tra cứu về cách soạn thảo (hình dưới).



8.13. Trong Turbo Pascal, muốn thực thi chương trình đến vị trí con trỏ tại sử dụng phím nào sau đây?

- A. Nhấn phím F4.
B. Nhấn phím F5.
C. Nhấn phím F6.
D. Nhấn phím F7.

8.14. Trong Turbo Pascal, để mở cửa sổ watch theo dõi giá trị các biến ta sử dụng tổ hợp phím nào?

- A. Nhấn tổ hợp phím Ctrl+F5.
B. Nhấn tổ hợp phím Ctrl+F6.
C. Nhấn tổ hợp phím Ctrl+F7.
D. Nhấn tổ hợp phím Ctrl+F8.

8.15. Trong Turbo Pascal, đang ở cửa sổ chương trình nguồn muốn xem lại màn hình output ta sử dụng tổ hợp phím nào?

- A. Nhấn tổ hợp phím Alt+F5.
B. Nhấn tổ hợp phím Alt+F6.
C. Nhấn tổ hợp phím Alt+F7.
D. Nhấn tổ hợp phím Alt+F8.

8.16. Trong Turbo Pascal, để đặt điểm gãy tại vị trí con trỏ ta sử dụng tổ hợp phím nào?

- A. Nhấn tổ hợp phím Ctrl+F5.
B. Nhấn tổ hợp phím Ctrl+F6.
C. Nhấn tổ hợp phím Ctrl+F7.
D. Nhấn tổ hợp phím Ctrl+F8.

8.17. Trong Turbo Pascal phát biểu nào dưới đây là đúng?

- A. Chỉ thực hiện một chương trình trên một file chương trình nguồn.
B. Có thể thực hiện nhiều chương trình trên một file chương trình nguồn.
C. Chỉ có thể thực hiện hai chương trình trên cùng một file chương trình nguồn.
D. Máy tính vẫn báo lỗi những lệnh sai sau lệnh "End." của chương trình.

II. Phần tự luận

8.1. Khi soạn thảo một chương trình cần tuân theo những nguyên tắc cơ bản nào?

8.2. Trong quá trình soạn thảo chương trình để tránh mất dữ liệu cần chú ý điều gì?

8.3. Khi chạy chương trình, nếu chương trình bị lặp vô hạn thì hãy sử dụng các tổ hợp phím nào để xử lý.

8.4. Soạn thảo một chương trình hiển thị dòng chữ ‘CHUC MUNG BAN DEN VOI TURBO PASCAL’ lưu thành file với tên vidu1 sau đó lưu lại bản thứ hai với tên mới là vidu2 và sửa câu hiển thị thành ‘CHUC THANH CONG’.

8.5. Tìm hiểu các chức năng trên Menu của môi trường soạn thảo Turbo Pascal.

Chương 3.

CẤU TRÚC Rẽ NHÁNH VÀ LẶP

Chương này giới thiệu các loại cấu trúc điều khiển trong lập trình đó là cấu trúc rẽ nhánh và cấu trúc lặp. Khái niệm bước đầu về lập trình có cấu trúc. Giới thiệu lệnh ghép Begin-End, lệnh rẽ nhánh If-Then, lệnh lặp For-Do và While-Do thể hiện các loại cấu trúc điều khiển trong ngôn ngữ lập trình Pascal.

Bài 9

CẤU TRÚC Rẽ NHÁNH

A. PHƯƠNG PHÁP GIẢI TOÁN

1. Cấu trúc rẽ nhánh.

+ Dạng thiếu:

Cấu trúc: If <điều kiện> then <câu lệnh>;

Trong đó: <điều kiện>: là một biểu thức logic.

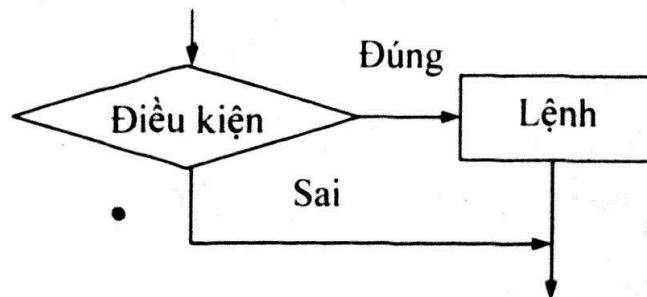
< câu lệnh>: là một lệnh nào đó của Pascal.

Sự thực hiện của máy:

+ Kiểm tra giá trị của <điều kiện>

+ Nếu <điều kiện> có giá trị đúng thì thực hiện <lệnh>

Sơ đồ:



+ Dạng đủ:

Cấu trúc: If <điều kiện> then <lệnh 1> else <lệnh 2>;

Trong đó <điều kiện>: là một biểu thức logic.

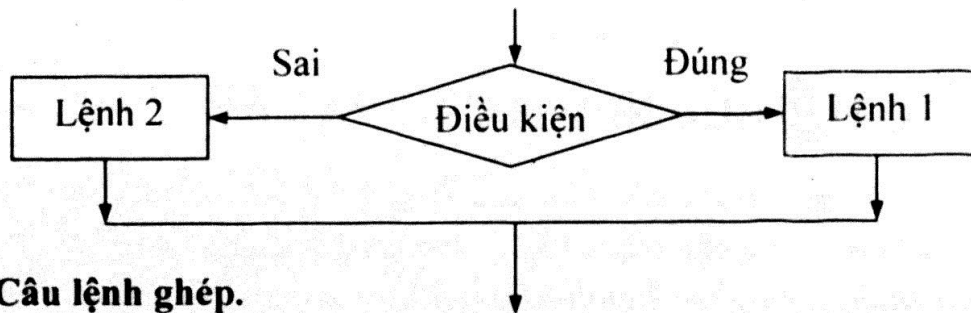
<lệnh 1>, <lệnh 2>: là các lệnh nào đó của Pascal.

Sự thực hiện của máy:

+ Kiểm tra giá trị của <điều kiện>

+ Nếu <điều kiện> có giá trị đúng thì thực hiện <lệnh 1>, ngược lại thì thực hiện <lệnh 2>.

Sơ đồ: Thực hiện theo sơ đồ sau:



1. Câu lệnh ghép.

Câu lệnh ghép có cú pháp:

Begin

<Danh sách các câu lệnh>;

End;

Trong đó <Danh sách các câu lệnh> là các câu lệnh của chương trình (từ hai câu lệnh trở lên).

Ví dụ

```
IF DELTA < 0 THEN WRITELN(' Phương trình vo nghiem.')
ELSE
    BEGIN
        X1 := (-B - SQRT(B*B - 4*A*C)) / (2*A);
        X2 := -B/A - X1;
    END;
```

3. Câu lệnh chọn CASE - OF

Xét bài toán : Lập trình đọc từ bàn phím tháng và năm. Tính và đưa ra màn hình số ngày của tháng.

Câu lệnh IF giải quyết việc rẽ nhánh tùy theo một trong hai khả năng. Bài toán trên liên quan tới việc chọn lựa một trong nhiều khả năng (rẽ nhiều nhánh). Khi đó, nếu dùng câu lệnh IF giải quyết sẽ làm chương trình rườm rà. Câu lệnh chọn **CASE - OF** trong TP cho phép rẽ nhiều nhánh một cách thuận lợi.

Câu lệnh **CASE - OF** có dạng :

```
CASE <Biểu thức nguyên, kí tự> OF
    <Danh sách 1>: <Câu lệnh 1>;
    <Danh sách 2>: <Câu lệnh 2>;
    . . . . .
    <Danh sách N>: <Câu lệnh N>
    [ ELSE <Câu lệnh N+1> ]
END;
```

Trong đó:

- *Danh sách I* là một hoặc nhiều giá trị của biểu thức nêu sau CASE, nếu có nhiều giá trị thì các giá trị viết cách nhau bởi dấu phẩy, $I = 1, 2, \dots, N$;
- Khi thực hiện, biểu thức sau từ khoá CASE sẽ được tính, giá trị nhận được sẽ lần lượt được kiểm tra xem nằm trong danh sách nào. Câu lệnh tương ứng với danh sách đầu tiên tìm thấy sẽ được thực hiện. Tiếp theo, thực hiện câu lệnh sau câu lệnh CASE - OF;
- Nếu giá trị tính được không xuất hiện ở bất kì danh sách nào thì câu lệnh sau ELSE (nếu có) sẽ được thực hiện.

Chương trình giải bài toán vừa nêu là như sau :

```
PROGRAM    Vi_du_CASE;
USES      CRT;
VAR    T, N, SN : INTEGER;
BEGIN
    CLRSCR;
    WRITE(' Cho biet thang va nam : ');
    READLN(T,N);
    CASE  T  OF
        4,6,9,11 : SN := 30;
        2 : IF (N MOD 400 = 0) OR
              ((N MOD 100 <> 0) AND (N MOD 4 = 0))
              THEN SN := 29 ELSE SN := 28
        ELSE SN := 31
    END;
    WRITELN('Thang',T, 'nam', N, ' co ', SN, ' ngay. ');
    READLN
END.
```

B. BÀI TẬP ÁP DỤNG

I. Bài tập trắc nghiệm

Hãy chọn phương án trả lời đúng nhất.

Phần cơ bản

9.1. Cú pháp biểu diễn cấu trúc rẽ nhánh dạng thiếu là:

- Iff <biểu thức> then <câu lệnh>;
- If <biểu thức> then <câu lệnh> else <câu lệnh>;

- C. If <biểu thức điều kiện> then < câu lệnh>;
- D. If <biểu thức điều kiện> then < câu lệnh> else <câu lệnh>;
- 9.2. Cú pháp biểu diễn cấu trúc rẽ nhánh dạng đủ là:**
- A. If <biểu thức> then < câu lệnh> else <câu lệnh>;
- B. If <biểu thức điều kiện> then < câu lệnh1> ;
else <biểu thức điều kiện> then < câu lệnh2>;
- C. If <biểu thức điều kiện> then < câu lệnh1> ;
else if <biểu thức điều kiện> then < câu lệnh2>;
- D. If <biểu thức điều kiện > then < câu lệnh1> else <câu lệnh2>
- 9.3. Câu lệnh ghép là:**
- A. Câu lệnh Begin và End.
- B. Từ hai câu lệnh trở lên.
- C. Câu lệnh có cấu trúc.
- D. Dãy các lệnh gồm từ hai câu lệnh trở lên và được đặt giữa từ khoá Begin và End;
- 9.4. Cú pháp If <biểu thức điều kiện> then < câu lệnh> có nghĩa là:**
- A. Nếu biểu thức điều kiện nhận giá trị đúng thì thực hiện câu lệnh.
- B. Nếu biểu thức điều kiện nhận giá trị sai thì thực hiện câu lệnh.
- C. Nếu biểu thức điều kiện nhận giá trị sai thì thực hiện câu lệnh ngược lại thực hiện câu lệnh khác.
- D. Đáp án B và C đều đúng.
- 9.5. Cú pháp If <biểu thức điều kiện> then < câu lệnh1> else <câu lệnh 2> có nghĩa là:**
- A. Nếu biểu thức điều kiện sai thì thực hiện câu lệnh 1 còn không sẽ thực hiện câu lệnh 2.
- B. Nếu biểu thức điều kiện đúng thì thực hiện câu lệnh 1 còn không sẽ thực hiện câu lệnh 2.
- C. Nếu biểu thức điều kiện đúng thì thực hiện câu lệnh 1 còn không sẽ thực hiện câu lệnh sau if.
- D. Nếu biểu thức điều kiện đúng thì thực hiện câu lệnh 2 còn không sẽ thực hiện câu lệnh 1.
- 9.6. Kết quả của đoạn chương trình sau là gì?**
- Var x: integer;
- Begin
- X:= 10;
- If (x - 3) > 5 then
- Writeln(' xin chuc mung');

End.

- A. Không hiển thị gì cả.
- B. In ra kết quả là 7.
- C. In ra màn hình câu: 'xin chuc mung'.
- D. In ra kết quả là 5.

9.7. Kết quả của đoạn chương trình sau là gì?

Var x,y: integer;

Begin

X:= 10; y:= 15;

If (x > y) then X:= x - y else y:= y- x

End.

- A. y = 5 và x=10
- B. X = 5 và y = 15
- C. X= 10 và y = 15.
- D. Chương trình báo lỗi.

9.8. Kết quả của đoạn chương trình sau là gì?

Var x,y: integer;

Begin

X:= 23; Y:= 45;

If (x > y) then write('so lon nhat la x');

If (y>x) then write('so lon nhat la y') else write(' hai so bang nhau');

End.

- A. So lon nhat la x
- B. So lon nhat la y
- C. Hai so bang nhau.
- D. Y lớn hơn x.

9.9. Kết quả của đoạn chương trình sau là gì?

Var x,y,z: integer;

Begin

X:= 10; Y:= 50;

If y/x >0 then z:= y/x;

If z > 2 then Write('pheap chia khac khong');

End.

- A. Z = 5 và xuất hiện thông báo: 'pheap chia khac khong'.
- B. Xuất hiện thông báo: 'pheap chia khac khong'.
- C. Z = 5, y=50 và xuất hiện thông báo: 'pheap chia khac khong'.
- D. X = 10, y = 50, z = 5 và thông báo: 'pheap chia khac khong'.

9.10. Cho đoạn chương trình:

Var x,y,z: integer;

Begin

X:= 10; Y:= 50; Z:= 5;

If x> y then z:= x - y

Else z:= y - x;

Write(z);

End.

Kết quả giá trị các biến x, y, z là:

A. X:= 10; Y:= 50; Z:= 5.

B. X:= 10; Y:= 50; Z:= -40.

C. X:= 10; Y:= 50; Z:= 40.

D. X:= 10; Y:= 50; Z:= 0.

9.11. Trong ngôn ngữ lập trình Pascal, câu lệnh nào sau đây là đúng?

A. if a=5 then

a:=d+1;

else

a:=d+2;

B. if a=5 then

a:=d+1

else

a:=d+2;

C. if a=5 then

a:=d+1

else

a:=d+2;

D. if a=5 then

a:=d+1

else

a:=d+2.

9.12. Trong ngôn ngữ lập trình Pascal, phát biểu nào sau đây là đúng với câu lệnh rẽ nhánh if...then...?

A. Nếu sau then muốn thực hiện nhiều câu lệnh thì các lệnh phải đặt giữa hai dấu ngoặc đơn.

B. Nếu sau then muốn thực hiện nhiều câu lệnh thì các lệnh phải đặt trước Begin và End;

C. Nếu sau then muốn thực hiện nhiều câu lệnh thì các lệnh phải đặt giữa Begin và End;

D. Nếu sau then muốn thực hiện nhiều câu lệnh thì các lệnh phải đặt giữa Begin và End.

9.13. Trong ngôn ngữ lập trình Pascal, phát biểu nào sau đây là đúng với câu lệnh rẽ nhánh if...then...else...?

A. Nếu sau then muốn thực hiện nhiều câu lệnh thì các lệnh phải đặt giữa hai dấu ngoặc đơn.

- B. Nếu sau then muốn thực hiện nhiều câu lệnh thì các lệnh phải đặt giữa Begin và End;
- C. Nếu sau then muốn thực hiện nhiều câu lệnh thì các lệnh phải đặt trước Begin và End;
- D. Nếu sau then muốn thực hiện nhiều câu lệnh thì các lệnh phải đặt giữa Begin và End.

9.14. Trong ngôn ngữ lập trình Pascal, phát biểu nào sau đây là đúng với câu lệnh rẽ nhánh if...then...else...?

- A. Nếu sau else muốn thực hiện nhiều câu lệnh thì các lệnh phải đặt giữa hai dấu ngoặc đơn.
- B. Nếu sau else muốn thực hiện nhiều câu lệnh thì các lệnh phải đặt trước Begin và End;
- C. Nếu sau else muốn thực hiện nhiều câu lệnh thì các lệnh phải đặt giữa Begin và End;
- D. Nếu sau else muốn thực hiện nhiều câu lệnh thì các lệnh phải đặt giữa Begin và End.

9.15. Trong ngôn ngữ lập trình Pascal, phát biểu nào sau đây là đúng?

- A. Sau mỗi câu lệnh đều phải có dấu chấm phẩy “;”
- B. Trước lệnh Else bắt buộc phải có dấu chấm phẩy “;”
- C. Có phân biệt chữ hoa và chữ thường.
- D. Câu lệnh trước câu lệnh End không nhất thiết phải có dấu chấm phẩy “;”.

9.16. Trong ngôn ngữ lập trình Pascal, muốn kiểm tra đồng thời cả ba giá trị của A, B, C có cùng lớn hơn 0 hay không ta viết câu lệnh If thế nào cho đúng?

- A. If A, B, C > 0 Then
- B. If (A > 0) And (B > 0) and (C > 0) Then
- C. If A > 0 And B > 0 and C > 0 Then
- D. If (A > 0) Or (B > 0) Or (C > 0) Then

II. Phần tự luận

Phần cơ bản

9.1. Viết chương trình giải phương trình bậc nhất $ax + b = 0$.

9.2. Viết chương trình giải phương trình bậc hai $ax^2 + bx + c = 0$.

9.3. Viết chương trình tìm giá trị lớn nhất, nhỏ nhất của hai số nhập vào từ bàn phím.

- 9.4. Viết chương trình nhập vào 3 số nguyên, in ra màn hình số lớn nhất, nhỏ nhất.
- 9.5. Viết chương trình nhập vào 3 số nguyên dương, kiểm tra xem chúng có phải là độ dài 3 cạnh của một tam giác hay không.

Phần nâng cao.

- 9.6. Giải và biện luận hệ phương trình tuyến tính:

$$\begin{cases} ax + by = m \\ cx + dy = n \end{cases}$$

- 9.7. Viết chương trình nhập vào một điểm kiểm tra từ bàn phím là một số nguyên dương và in ra kết quả xếp loại theo tiêu chí sau: Nếu điểm dưới 5 thì xếp loại yếu, nếu điểm là 5 hoặc 6 thì xếp loại trung bình, nếu điểm là 7 hoặc 8 thì xếp loại khá, nếu điểm là 9 hoặc 10 thì xếp loại giỏi.
- 9.8. Viết chương trình giải phương trình trùng phương: $ax^4 + bx^2 + c = 0$.
- 9.9. Viết chương trình nhập vào 3 số nguyên dương từ bàn phím và in ra màn hình 3 số đó theo trật tự tăng dần.
- 9.10. Lập chương trình tính diện tích và chu vi của 1 hình tròn bán kính R.
- 9.11. Viết chương trình nhập vào hai số nguyên dương sau đó tìm ước chung lớn nhất và bội chung nhỏ nhất của chúng.
- 9.12. Viết chương trình nhập vào số KW điện tiêu thụ của mỗi hộ gia đình rồi tính số tiền phải trả biết rằng giá tiền được tính như sau:
- Từ KW thứ 1 đến 100 được tính giá 1000 đồng.
 - Từ KW thứ 101 đến 200 được tính giá 3000 đồng.
 - Từ KW thứ 201 trở lên tính giá 5000 đồng.

Bài 10

CÁU TRÚC LẶP

A. PHƯƠNG PHÁP GIẢI TOÁN

1. Cấu trúc lặp với số lần lặp biết trước.

Lệnh lặp For của ngôn ngữ lập trình Pascal có hai dạng:

- Dạng tiến:

Cấu trúc:

For <biến đếm>:=<Giá trị đầu> To <Giá trị cuối> Do <lệnh cần lặp>;

Biến đếm: là biến kiểu nguyên, kí tự hoặc miền con.

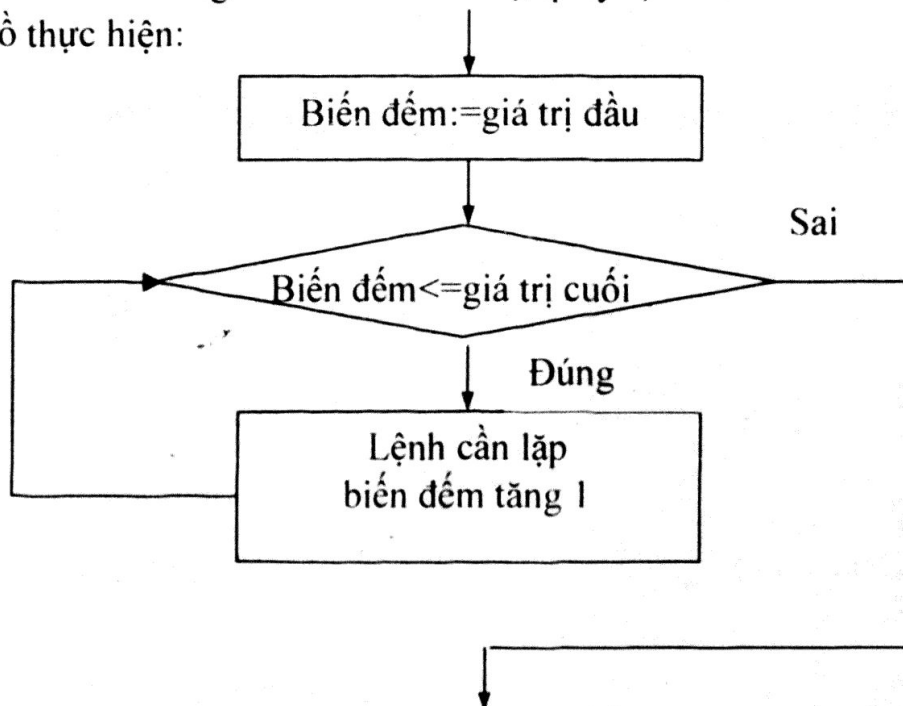
Giá trị đầu, giá trị cuối là biểu thức cùng kiểu với biến đếm. Giá trị đầu phải nhỏ hơn hoặc bằng giá trị cuối.

Sự thực hiện của máy:

Bước 1: Tính giá trị đầu, gán cho biến đếm.

Bước 2: Nếu biến đếm $\leq$ giá trị cuối thì
thực hiện lệnh cần lặp
tăng biến đếm 1 đơn vị, quay lại bước 2

Sơ đồ thực hiện:



- Dạng lùi:

Cấu trúc: For <biến đếm>:=<Giá trị cuối> Downto <Giá trị đầu> Do <lệnh cần lặp>;

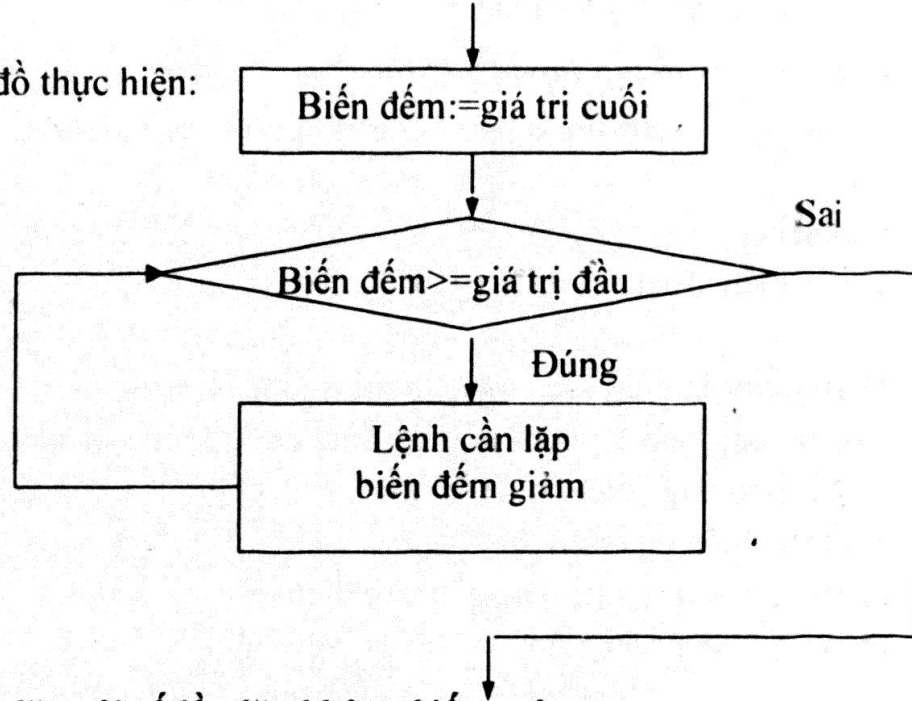
Giá trị đầu phải nhỏ hơn hoặc bằng giá trị cuối.

Sự thực hiện của máy:

Bước 1: Tính giá trị cuối, gán cho biến đếm.

Bước 2: Nếu biến đếm $\geq$ giá trị đầu thì
thực hiện lệnh cần lặp
giảm biến đếm 1 đơn vị, quay lại bước 2

Sơ đồ thực hiện:



2. Cấu trúc lặp với số lần lặp không biết trước.

- Cấu trúc While ...do.

Cấu trúc: **While** <điều kiện> **Do** <lệnh cần lặp>;

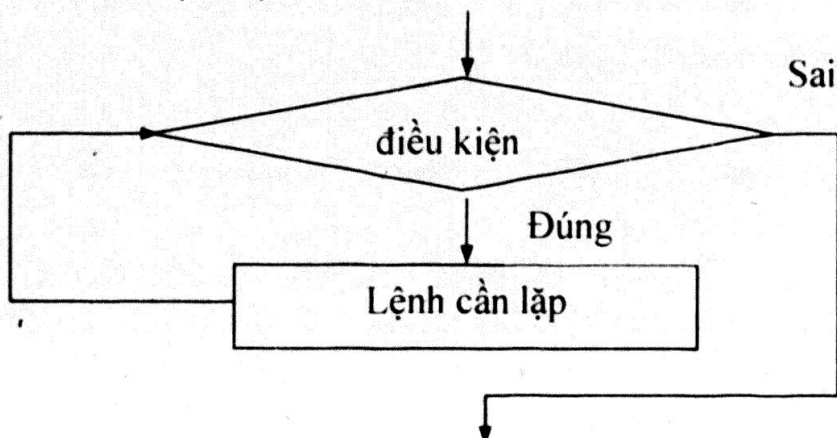
Điều kiện: là biểu thức quan hệ hoặc biểu thức logic.

Sự thực hiện của máy:

Bước 1: Tính giá trị của <điều kiện>.

Bước 2: Nếu <điều kiện> có giá trị đúng thì:
thực hiện lệnh cần lặp, quay lại bước 1

Sơ đồ thực hiện:



Cấu trúc lặp Repeat .. Until:

- Cú pháp :

REPEAT

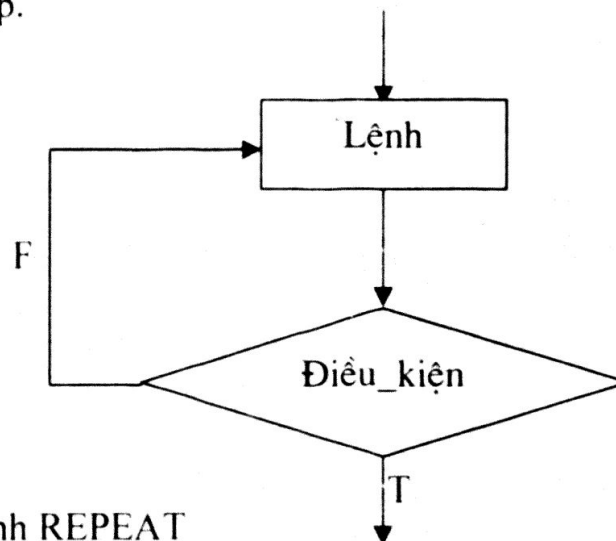
<lệnh>

UNTIL <điều kiện>;

Các câu lệnh giữa REPEAT và UNTIL tạo nên 1 danh sách câu lệnh gọi là thân REPEAT.

Vì hai từ khoá REPEAT và UNTIL cho một giới hạn tự nhiên các câu lệnh thân REPEAT nên có thể bỏ từ khoá BEGIN và END; nếu thân REPEAT là câu lệnh ghép.

- Sơ đồ khối:



- Sự thực hiện của câu lệnh REPEAT

Bước 1: Thực hiện tuần tự các câu lệnh giữa REPEAT và UNTIL.

Bước 2: Sau khi thực hiện câu lệnh cuối cùng trong danh sách lệnh, tính giá trị của điều kiện, nếu kết quả là FALSE thì quay lại bước 1, ngược lại kết thúc vòng lặp REPEAT.

Sự thực hiện của câu lệnh lặp Repeat có thể được phát biểu như sau: Lặp lại thực hiện <lệnh> cho đến khi nào giá trị của biểu thức điều kiện là đúng.

Như vậy <Lệnh> được thực hiện trước khi kiểm tra <BTĐK> do đó lệnh luôn được thực hiện ít nhất 1 lần.

Ví dụ: Tính số tháng để có được số tiền tiết kiệm b, tiền gốc là a gửi sau t tháng với lãi suất $k=0.01$.

```
PROGRAM Tietkiem;
```

```
CONST k=0.01;
```

```
VAR a, b : REAL;
```

```
t : INTEGER;
```

```
BEGIN
```

```
WRITE(' So tien gui ban dau la: '); READLN(a);
```

```
WRITE(' So tien muon dat duoc la: '); READLN(b);
```

```

t := 0;
REPEAT
    a := a + a*k;
    t := t + 1;
UNTIL a>=b
WRITELN(' Can gui trong ', t, ' thang. ');
READLN;
END.

```

- Chú ý:

+ Đề tránh trường hợp lặp vô hạn trong thân vòng lặp phải có lệnh thay đổi giá trị của <BTĐK>.

+ Các lệnh WHILE, REPEAT có thể lồng nhau nhưng chú ý không được chéo nhau.

B. BÀI TẬP ÁP DỤNG

I. Câu hỏi trắc nghiệm

Hãy chọn phương án trả lời đúng nhất.

Phần cơ bản.

10.1. Có mấy dạng cấu trúc lặp?

- | | |
|------------|------------|
| A. 1 dạng. | C. 3 dạng. |
| B. 2 dạng. | D. 4 dạng. |

10.2. Cấu trúc lặp với số lần biết trước là cấu trúc:

- | | |
|-----------------|--------------|
| A. Repeat Until | C. For to do |
| B. While Do | D. For do |

10.3. Cấu trúc lặp với số lần không biết trước là cấu trúc:

- | | |
|------------------|-------------------|
| A. For Downto do | C. Repeat Until |
| B. While Do | D. Đáp án B và C. |

10.4. Cú pháp khai báo cấu trúc lặp dạng tiến với số lần lặp biết trước là:

- A. For <biến đếm>:=<giá trị đầu> to <giá trị cuối> <Câu lệnh>;
 B. For <biến đếm>:=<giá trị đầu> to <giá trị cuối> do
 C. For <biến đếm>:=<giá trị đầu> to <giá trị cuối> do <Câu lệnh>;
 D. Không có đáp án nào đúng.

10.5. Cú pháp khai báo cấu trúc lặp dạng lùi với số lần lặp biết trước là:

- A. For <biến đếm>:=<giá trị đầu> downto <giá trị cuối> do <Câu lệnh>;
 B. For <biến đếm>:=<giá trị đầu> downto <giá trị cuối> <câu lệnh>;
 C. For <biến đếm>:=<giá trị đầu> to <giá trị cuối> do <Câu lệnh>;
 D. For <biến đếm>:=<giá trị cuối> downto <giá trị đầu> do <Câu lệnh>;

10.6. Trong các phát biểu sau, phát biểu nào sai?

- A. Trong cấu trúc lặp For biến đếm là biến có kiểu nguyên.
- B. Để cấu trúc lặp For dạng tiến có ý nghĩa thì giá trị đầu phải nhỏ hơn hoặc bằng giá trị cuối.
- C. Trong quá trình thực hiện vòng lặp for tiến thì biến đếm tự động tăng lên một đơn vị sau mỗi vòng lặp for.
- D. Cần phải thay đổi giá trị của biến đếm trong vòng lặp for.

10.7. Cấu trúc lặp với số lần lặp không xác định Repeat ... Until có cú pháp là:

A. Repeat <biểu thức điều kiện>

 <Câu lệnh 1>;

 <Câu lệnh 2>;

 <Câu lệnh n>;

Until <Biểu thức logic>;

B. Repeat

 <Câu lệnh 1>;

 <Câu lệnh 2>;

 <Câu lệnh n>;

Until <Biểu thức logic>;

C. Repeat

 <Câu lệnh 1>;

 <Câu lệnh 2>;

 <Câu lệnh n>;

Until

 <Câu lệnh n+1>;

D. Repeat <Biểu thức logic> Until

 <Câu lệnh 1>;

 <Câu lệnh 2>;

 <Câu lệnh n>;

10.8. Cấu trúc lặp với số lần lặp không xác định While ... do có cú pháp là:

A. While <biểu thức logic> do <Câu lệnh>;

B. Write <biểu thức> do <Câu lệnh>;

C. While <biểu thức> do <Câu lệnh1> Else <Câu lệnh 2>;

D. While <Câu lệnh> do <biểu thức logic>;

10.9. Trong các phát biểu sau phát biểu nào sai:

- A. Các câu lệnh nằm giữa Repeat ...Until không cần phải sử dụng cặp từ khoá Begin...End;
- B. Trong vòng lặp Repeat thường phải có câu lệnh làm thay đổi giá trị của biểu thức logic nếu không sẽ xảy ra tình trạng lặp vô hạn.
- C. Trong vòng lặp Repeat ...Until các câu lệnh phải thực hiện ít nhất một lần.
- D. Trong vòng lặp Repeat ...Until phải cần sử dụng cặp từ khoá Begin...End;

10.10. Trong vòng lặp While <Biểu thức điều kiện> do <câu lệnh>, phát biểu nào dưới đây đúng?

- A. Chừng nào biểu thức điều kiện còn sai thì thực hiện câu lệnh.
- B. Trong vòng lặp các câu lệnh thực hiện ít nhất là 1 lần.
- C. Chừng nào biểu thức điều kiện còn đúng thì thực hiện câu lệnh.
- D. Các câu lệnh trong vòng lặp không phải nằm giữa từ khoá Begin...End;

Phân nâng cao.

10.11. Trong vòng lặp While <Biểu thức điều kiện> do <câu lệnh>, phát biểu nào dưới đây đúng:

- A. Các lệnh trong vòng lặp có thể không được thực hiện một lần nào cả.
- B. Không được làm thay đổi biểu thức logic của vòng lặp, nếu không vòng lặp sẽ lặp vô hạn.
- C. Cách thực hiện của vòng lặp là giống cách thực hiện của vòng lặp Repeat...Until.
- D. Biểu thức điều kiện của vòng lặp phải là biểu thức số học.

10.12. Đoạn chương trình sau làm gì?

For i:= 1 to 10 do Write(i:2);

- A. Hiển thị các số: 1 3 5 7 9.
- B. Hiển thị các số: 2 4 6 8 10.
- C. Hiển thị các số: 1 2 3 4 5 6 7 8 9 10.
- D. Hiển thị các số: 10 9 8 7 6 5 4 3 2 1

10.13. Đoạn chương trình sau làm gì?

For i:= 4 to 10 do Write(i*2:2);

- A. Hiển thị các số: 4 5 6 7 8 9 10.
- B. Hiển thị các số: 10 9 8 7 6 5 4.
- C. Hiển thị các số: 20 18 16 14 12 10 8.
- D. Hiển thị các số: 8 10 12 14 16 18 20.

10.14. Cho đoạn chương trình:

Var i,s: Integer;

Begin

S:= 0;

For i:= 5 downto 1 do s:= s+i;

Write('Tổng là:', S);

A. Tổng là: 15.

B. Tổng là: 10.

C. Tổng là: 20.

D. Tổng là: 5.

10.15. Cho đoạn chương trình:

Var i,j,s: Integer;

Begin

For i:= 1 to 3 do Write(i:2);

For j:= 4 downto 2 do Write(j:2);

End;

Kết quả khi chạy chương trình là:

A. 1 2 3 2 3 4

B. 1 2 3 4 3 2

C. 2 3 4 1 2 3

D. 3 2 1 4 3 2

10.16. Đoạn chương trình sau làm gì?

Repeat

Write(' Nhập giá trị n:');

Readln(n);

Until n > 0;

A. Kiểm tra số n dương hay không.

B. Nhập số n vào.

C. Nhập giá trị cho n cho đến khi nào giá trị n nhập vào > 0.

D. Nhập giá trị cho n cho đến khi nào n < 0.

10.17. Đoạn chương trình dưới đây làm gì?

Var mk:string[6];

Begin

Repeat

Write(' nhập vào mật khẩu');

Readln(mk);

Until mk= 'phuong';

Writeln(' Ban da vao dung mat khau');
End.

- A. Chương trình thoát khi nhập vào chuỗi PHUONG.
- B. Chương trình thoát khi nhập vào chuỗi 'phuong' và hiển thị dòng chữ 'Ban da vao dung mat khau'.
- C. Chương trình hiển thị dòng chữ: 'Ban da vao dung mat khau'.
- D. Chương trình nhập giá trị cho biến mk.

10.18. Đoạn chương trình sau làm gì:

```
Var n: word;  
Begin  
N:=5;  
Repeat  
  N:= n-1;  
Until n = 0;  
Write(n);  
End.
```

- A. Cho kết quả $n = 5$.
- B. Cho kết quả $n = 4, n = 3, n = 2, n = 1$.
- C. Cho kết quả $n = 0$.
- D. Cho kết quả $n = 1, n = 2, n = 3, n = 4, n = 5$.

10.19. Cho đoạn chương trình:

```
Var a,b: byte; ch: char;  
Begin  
  Repeat  
    Write(' nhap gia tri cho a va b');  
    Readln(a,b);  
    Writeln(' tong la',a+b);  
    Writeln('Ban co tiep tục nua không');  
    Readln(ch);  
  Until Uppcase(ch) = 'K' ;  
End.
```

- A. Chương trình tính tổng của hai số a và b nhập vào từ bàn phím, mỗi lần tính xong xuất hiện thông báo: 'Ban co tiep tục nua không'. Nếu nhập c thì tiếp tục, nhập k thì kết thúc.
- B. Chương trình nhập hai số a và b từ bàn phím, mỗi lần nhập xong xuất hiện thông báo: 'Ban co tiep tục nua không'. Nếu nhập 'c' thì tiếp tục, nhập 'k' thì kết thúc.

- C. Chương trình xuất hiện thông báo: 'Bạn có tiếp tục nữa không'.
Nếu nhập 'c' thì tiếp tục, nhập 'k' thì kết thúc.
- D. Chương trình yêu cầu nhập vào kí tự 'c' hoặc 'k', nếu nhập 'c' thì tiếp tục, nhập 'k' thì kết thúc.

10.20. Cho đoạn chương trình

Var n: byte;

Begin

N:= 10;

While n > 2 do

N:= n - 2;

Write(n+2);

End.

Kết quả khi chạy chương trình là:

A. n = 0.

C. n = 4.

B. n = 2.

D. n = 6.

10.21. Trong ngôn ngữ lập trình Pascal, về mặt cú pháp câu lệnh nào sau đây là đúng với cấu trúc lặp For có một lệnh con?

A. For i:=1 to 100 do

C. For i:=1 to 100 do

a:=a -1;

a:=a -1

B. For i:=1 to 100 do;

D. For i:=1; to 100 do

a:=a -1;

a:=a -1;

10.22. Trong ngôn ngữ lập trình Pascal, về mặt cú pháp câu lệnh nào sau đây là đúng với cấu trúc lặp For có nhiều lệnh con?

A. For i:=1 to 100 do

C. For i:=1 to 100 do

a:=a -1;

Begin

b:=a-c;

a:=a -1;

End For;

b:=a-c

End;

B. For i:=1 to 100 do ;

D. For i:=1 to 100 do

Begin

a:=a -1

a:=a -1;

b:=a-c;

b:=a-c;

End;

10.23. Trong ngôn ngữ lập trình Pascal, về mặt cú pháp câu lệnh nào sau đây là đúng với cấu trúc lặp While có một lệnh con?

A. While a>5 do

C. While a>5 do

a:=a -1;

a:=a -1

B. While a>5 do ;
a:=a -1;

D. While a>5; do
a:=a -1;

10.24. Trong ngôn ngữ lập trình Pascal, về mặt cú pháp câu lệnh nào sau đây là đúng với cấu trúc lặp While có một lệnh con?

A. While a>5 and a<17 do
a:=a -1;

C. While (a>5)and(a<17) do
a:=a -1

B. While (a>5)and(a<17) do ;
a:=a -1;

D. While (a>5)and(a<17) do
a:=a -1;

10.25. Trong ngôn ngữ lập trình Pascal, về mặt cú pháp câu lệnh nào sau đây là đúng với cấu trúc lặp While có nhiều lệnh con?

A. While a>5 do
a:=a -1;
b:=a-c;

C. While a>5 do
a:=a -1
b:=a-c;

EnWhile;
B. While a>5 do;
Begin
a:=a -1;
b:=a-c;
End;

D. While a>5 do
Begin
a:=a -1;
b:=a-c
End;

10.26. Trong ngôn ngữ lập trình Pascal, ta có câu lệnh lặp For i := 1 to 100 do Write(i:4); Nếu muốn sử dụng vòng lặp While ... Do thực hiện công việc tương tự như vậy thì ta phải viết câu lệnh While ... Do như thế nào?

A. I := 1;
While I <= 100 Do
Begin
Write(i: 4);
I := I + 1;
End;

C. I := 100;
While I > 1 Do
Begin
I := I - 1;
Write(i: 4);
End;

B. I := 0;
While I < 100 Do
Begin
I := I + 1;
Write(i: 4);
End;

D. Cả ba cách trên đều đúng.

10.27. Trong ngôn ngữ lập trình Pascal, đoạn chương trình sau đưa ra màn hình kết quả gì?

For i := 10 to 1 do Write(i, ' ')

A. 1 2 3 4 5 6 7 8 9 10

B. 10 9 8 7 6 5 4 3 2 1

C. Đưa ra 10 dấu cách

D. Không đưa ra kết quả gì

10.28. Trong ngôn ngữ lập trình Pascal, đoạn chương trình sau cho kết quả là gì?

$T := 0;$

For $i := 1$ to n do If $(i \bmod 3 = 0)$ Then $T := T + i*i;$

A. Tính tổng các số nguyên trong phạm vi từ 1 đến n ;

B. Tính tổng các số chia hết cho 3 trong phạm vi từ 1 đến n ;

C. Tính tổng bình phương các số chia hết cho 3 trong phạm vi từ 1 đến n ;

D. Tính tổng bình phương các số trong phạm vi từ 3 đến n .

10.29. Trong ngôn ngữ lập trình Pascal, đoạn chương trình sau thực hiện công việc gì?

$i := 0;$ While $i <> 0$ Do Write($i, ' '$);

A. Đưa ra màn hình 10 chữ số 0

B. Không đưa ra thông tin gì

C. Lặp vô hạn việc đưa ra màn hình số 0

D. Đưa ra màn hình một chữ số 0

10.30. Trong ngôn ngữ lập trình Pascal, đoạn chương trình sau cho kết quả trên màn hình là gì với i là biến số nguyên?

For $i := 1$ to 100 do If $i \bmod 9 = 0$ Then Write($i, ' '$);

A. 1 2 3 4 5 6 ... 100

B. 9 18 27 36 45 54 63 72 81 90 99

C. 9 18 27 36 45 54 63 72 81 90 99

D. 9 9 9 9 9 9 9 9

10.31. Trong ngôn ngữ lập trình Pascal, đoạn chương trình sau thực hiện công việc gì?

$T := 0;$

For $i := 1$ To N Do

If $(i \bmod 3 = 0)$ or $(i \bmod 5 = 0)$ Then $T := T + i;$

A. Tính tổng các số chia hết cho 3 hoặc 5 trong phạm vi từ 1 đến N

B. Tính tổng các ước thực sự của N

C. Tính tổng các số chia hết cho cả 3 và 5 trong phạm vi từ 1 đến N

D. Tìm một ước của số N

10.32. Trong ngôn ngữ lập trình Pascal, hãy cho biết giá trị của M sau khi thực hiện đoạn chương trình sau với $a=10$ và $b=12$?

$M := a;$

If $a < b$ Then $M := b;$

A. $M = 12$

B. $M = 10$

C. M nhận cả hai giá trị trên

D. M không nhận giá trị nào

10.33. Trong ngôn ngữ lập trình Pascal, hãy cho biết đoạn chương trình sau làm công việc gì?

$I := 0; T := 0;$

While $I < 10000$ do

Begin

$T := T + i;$

$I := I + 2;$

End;

A. Tính tổng các số tự nhiên nhỏ hơn hoặc bằng 10000

B. Tính tổng các số tự nhiên chẵn nhỏ hơn 10000

C. Tính tổng các số tự nhiên chẵn nhỏ hơn hoặc bằng 10000

D. Tính tổng các số tự nhiên nhỏ hơn 10000

II. Phần tự luận

Phần cơ bản

10.1. Viết chương trình tính các tổng sau:

a. $S = 1 + 2 + 3 + \dots + n.$

b. $S = 1 - 2 + 3 - \dots + (-1)^{n-1} n.$

c. $S = 1 + 1*2 + 1*2*3 + \dots + 1*2*3 \dots *n.$

d. $S = 1 - 1*2^2 + 1*2*3^2 - \dots + (-1)^{n-1} 1*2*3 \dots *n^2.$

10.2. Viết chương trình tính tổng các số chẵn từ 1 đến 100.

10.3. Viết chương trình tính tổng các số lẻ từ 1 đến 100.

10.4. Viết chương trình tìm ước số chung lớn nhất, bội số chung nhỏ nhất của hai số nguyên dương.

10.5. Viết chương trình nhập vào một số nguyên dương rồi in ra màn hình số đảo ngược của số đó.

10.6. Viết chương trình nhập vào hai số nguyên dương m, n ($m < n$) cho biết tất cả các số hoàn hảo nằm trong đoạn $[m, n]$. Cho biết số hoàn hảo là số bằng tổng tất cả các ước số của nó trừ số đó, ví dụ: số 6.

10.7. Viết chương trình kiểm tra xem một số nào đó có phải là số nguyên tố hay không?

10.8. Viết chương trình tìm tất cả các số nguyên tố nằm trong đoạn $[1, n]$ với n nhập từ bàn phím.

10.9. Tìm số của dãy Fibonacci lớn nhất thỏa mãn bé hơn 2000. Dãy Fibonacci là dãy $a_1 = 1, a_2 = 1, a_n = a_{n-1} + a_{n-2}$ ($n \geq 3$)

10.10. Viết chương trình phân tích một số nguyên thành tích của các số nguyên tố

Phần nâng cao.

10.11. Viết chương trình giải bài toán:

*“Vừa gà vừa chó
Bỏ lại cho tròn
Ba mươi sáu con
Một trăm chân chẵn”.*

Hỏi có bao nhiêu con gà, bao nhiêu con chó?

10.12. Một người có số tiền là t đem đi gửi ở ngân hàng với lãi suất k /tháng, cho biết lãi của tháng trước được nhập vào tiền gốc để tính lãi cho tháng tiếp theo. Hỏi để có được số tiền là $\max$ ($\max > t$) thì người đó phải gửi tiết kiệm bao nhiêu tháng. Yêu cầu $t, k, \max$ được nhập vào từ bàn phím.

10.13. Dùng vòng lặp For để viết các số từ 0 đến 99 theo cách sau:

0 1 2 3 4 5 6 7 8 9
10 11 12 13 14 15 16 17 18 19

.....

90 91 92 93 94 95 96 97 98 99

10.14. Tính giai thừa theo công thức:

$$n! = \begin{cases} 1 & \text{nếu } n = 0; \\ 1.2.3 \dots n & \text{nếu } n > 0. \end{cases}$$

10.15. Viết chương trình hiển thị ra bốn dòng các chữ cái hoa và thường theo thứ tự từ ‘a’ đến ‘z’ ; ‘A’ đến ‘Z’ và ngược lại ‘z’ xuống ‘a’; ‘Z’ xuống ‘A’. Với quy cách mỗi kí tự chiếm 2 chỗ.

10.16. Viết chương trình tính $e^x = 1 + x / 1! + x^2 / 2! + \dots + x^n / n!$

10.17. Viết chương trình tính $\cos(x)$ theo chuỗi Taylor.

$$\cos x = 1 - x^2/2! + x^4/4! - x^6/6! + \dots$$

10.18. Viết chương trình in ra bảng mã ASCII lên màn hình.

10.19. Viết chương trình tính tổng n số nguyên dương đầu tiên bằng hai cách: dùng vòng lặp Repeat và While.

10.20. Viết chương trình tìm tất cả các số có 3 chữ số abc sao cho tổng các lập phương của các chữ số thì bằng chính nó. Nghĩa là:

$$abc = 100a + 10b + c = a^3 + b^3 + c^3.$$

Chương 4

Kiểu dữ liệu có cấu trúc

Chương này trình bày ba kiểu dữ liệu có cấu trúc quan trọng:

- Kiểu dữ liệu mảng.
- Kiểu xâu kí tự.
- Kiểu bản ghi.

Bài 11

Kiểu mảng

A. PHƯƠNG PHÁP GIẢI TOÁN

Mảng là kiểu dữ liệu có cấu trúc bao gồm một số cố định các phần tử có cùng kiểu, cùng một tên chung. Mỗi phần tử của mảng có thể xem như là một biến.

1.-Mảng một chiều

- Mảng một chiều là dãy hữu hạn các phần tử có cùng kiểu dữ liệu. Các phần tử trong mảng có cùng chung một tên và phân biệt nhau bởi chỉ số. Để mô tả mảng một chiều cần xác định được kiểu của các phần tử và cách đánh số các phần tử của nó.

- Hầu hết các ngôn ngữ lập trình đều có quy tắc cho phép xác định: tên kiểu mảng, số lượng phần tử, kiểu dữ liệu của từng phần tử, cách khai báo một biến mảng và cách tham chiếu đến từng phần tử của mảng.

- Cách tạo kiểu mảng một chiều và cách khai báo biến.

TYPE tên_kiểu_mảng = Array[kiểu_chi_số] Of kiểu_thành_phần;

VAR tên_biến: tên_kiểu_mảng;

- Có thể khai báo mảng trực tiếp như sau:

VAR tên_biến : Array[kiểu_chi_số] Of kiểu_thành_phần;

Trong đó:

+ Kiểu_chi_số: thường là một đoạn số nguyên (hoặc đoạn kí tự) liên tục, có dạng n1..n2 với n1 và n2 là các biểu thức nguyên (hoặc kí tự) xác định chỉ số đầu và chỉ số cuối của mảng.

+ Kiểu_thành_phần: là kiểu dữ liệu chung của mọi phần tử trong mảng.

- Tham chiếu đến từng phần tử của mảng: Tên_biến[chỉ số]

Các giải thuật trên mảng một chiều:

Giả sử đã khai báo mảng a, b có n phần tử, biến chỉ số i, có kiểu phần tử là kiểu nguyên. Dưới đây là một số giải thuật:

+ Gán một biến mảng cho một biến mảng khác:

For i:= 1 to n do

a[i]:= i;

b:=a;

+ Nhập các phần tử cho mảng:

For i:= 1 to n do

Begin

Write('a[',i, ']=');

Readln(a[i]);

End;

+ Hiện thị các phần tử của mảng:

For i:= 1 to n do

Begin

Write('a[',i, ']= ', a[i]);

Writeln;

End;

+ Sắp xếp các phần tử của mảng:

For i:= 1 to n-1 do

For j:= i+1 to n do

If a[i] > a[j] then

Begin

Tg:= a[i];

a[i]:= a[j];

a[j]:=tg;

End;

+ Tìm các phần tử của mảng bằng một giá trị cho trước (giả sử đã khai báo các biến)

Write('nhap gia tri so sanh'); Readln(x);

dem:=0;

For i:= 1 to n do

If a[i] = x then

Dem:=dem+1;

Ví dụ:

Input : Số nguyên dương N ($N < 250$) và dãy N số nguyên dương $a_1, a_2, \dots, a_n$, mỗi số đều không vượt quá 500.

Output : Số lượng giá trị khác nhau của dãy số và chỉ số của các phần tử của dãy số ứng với mỗi giá trị.

Để viết chương trình giải bài toán này, ta cần khai báo:

- Biến mảng A để nhập các phần tử của dãy số,
- Biến mảng Dem , $Dem[i]$ để đếm số phần tử của dãy có giá trị bằng i .

Từ hạn chế của đầu bài, suy ra giá trị của mỗi phần tử của dãy số nằm trong phạm vi từ 1 đến 500 và số lượng phần tử của dãy có giá trị i là không quá 250, nên mảng Dem có chỉ số chạy từ 1 đến 500 và mỗi thành phần của nó có kiểu byte.

Uses CRT;

Const

 Nmax = 250;

 Giatrimax = 500;

Type

 ArrInt = Array[1..Nmax] Of Integer;

 ArrByte = Array[1..Giatrimax] Of Byte;

Var

 N, I, J, Soluong : Integer;

 A : ArrInt;

 Dem : ArrByte;

Begin

 Clrscr;

 Write('Nhap so phan tu, N = '); Readln(N);

 For i:=1 to N Do

 Begin

 Write('phan tu thu ', i, ' bang ');

 Readln(A[i]);

 End;

 For i:=1 to Giatrimax Do dem[i] := 0;

 For i:=1 to N Do inc(dem[A[i]]);

 Soluong:=0;

 For i:=1 to N Do

```

    if dem[i]>0 then inc(Soluong);
Writeln('co tat ca ',Soluong,' gia tri khac nhau');
For i:=1 to giatrimax Do
    if dem[i]<>0 then
        Begin
            Write('giatri',i,'có',dem[i],'phan tu:');
            For J:=1 To N Do
                If A[j]=i Then Write(J:3);Writeln;
        End;
Readln
End.

```

2. Mảng hai chiều.

- Khái niệm: Mảng hai chiều là một bảng các phần tử có cùng kiểu dữ liệu, mảng hai chiều có hai tập chỉ số đó là chỉ số dòng và chỉ số cột.

- Hầu hết các ngôn ngữ lập trình đều có quy tắc cho phép xác định: tên kiểu mảng hai chiều, số lượng phần tử của mỗi chiều, kiểu dữ liệu của từng phần tử, cách khai báo một biến mảng và cách tham chiếu đến từng phần tử của mảng.

- Cách tạo kiểu mảng và cách khai báo biến.

- Tạo kiểu dữ liệu mảng hai chiều:

TYPE tên_kiểu_mảng = Array[kiểu_chi_số_dòng, kiểu_chi_số_cột]
Of kiểu_thành_phần;

- Khai báo biến mảng một chiều: VAR tên_biến:tên_kiểu_mảng;

- Có thể khai báo biến mảng trực tiếp:

VAR tên_biến : Array[kiểu_chi_số_dòng, kiểu_chi_số_cột] Of
kiểu_thành_phần;

- Tham chiếu đến từng phần tử: Tên_biến[chỉ_số_dòng, chỉ_số_cột]

Trong đó:

+ Kiểu_chi_số: thường là một đoạn số nguyên (hoặc đoạn kí tự) liên tục, có dạng n1..n2 với n1 và n2 là các biểu thức nguyên (hoặc kí tự) xác định chỉ số đầu và chỉ số cuối của mảng.

+ Kiểu_thành_phần: là kiểu dữ liệu chung của mọi phần tử trong mảng.

+ Các giải thuật trên mảng một chiều:

Giả sử đã khai báo mảng a, b là hai mảng hai chiều, biến chỉ số i, j có kiểu phần tử là kiểu nguyên. Dưới đây là một số giải thuật:

+ Gán một biến mảng cho một biến mảng khác:

```

For i:= 1 to n do
  For j:= 1 to m do
    a[i,j]:= i;
    b:=a;
  + Nhập các phần tử cho mảng:
  For i:= 1 to n do
    For j:= 1 to m do
      Begin
        Write('a[',i,j,']=');
        Readln(a[i,j]);
      End;
    End;
  + Hiện thị các phần tử của mảng:
  For i:= 1 to n do
    For j:= 1 to m do
      Begin
        Write('a[',i,j,']= ', a[i,j]);
        Writeln;
      End;
  + Tìm các phần tử của mảng bằng một giá trị cho trước (giả sử đã
  khai báo các biến)
  Write('nhap gia tri so sanh'); Readln(x);
  dem:=0;
  For i:= 1 to n do
    For j:= 1 to m do
      If a[i,j] = x then
        Dem:=dem+1;

```

Ví dụ:

Xây dựng mảng hai chiều B gồm 10 dòng, 10 cột với các phần tử là các số nguyên được tạo ngẫu nhiên trong phạm vi từ -20 đến 20. Đưa ra màn hình tất cả các phần tử của mảng có giá trị lớn hơn số k cho trước. Giá trị của k được nhập từ bàn phím.

```
Program MangHaiChieu;
```

```
Uses CRT;
```

```
Var
```

```
  b : array[1..10, 1..10] of integer;
```

```

i, j, k : integer;
Begin
  Clrscr;
  Randomize; { khai dong so ngau nhien}
  For i:= 1 to 10 do
    begin
      for j:= 1 to 10 do
        begin
          b[i,j]:= random(41)-20;
          write(b[i,j], ' ');
        end;
      writeln;
    end;
  write('Nhap vao gia tri k = '); readln(k);
  write('Cac phan tu mang co gia tri lon hon ',k,' ');
  For i:= 1 to 10 do
    For j:= 1 to 10 do
      If b[i, j]>k then write (b[i, j], ' ');
    Readln
  End.

```

Trong chương trình trên ta sử dụng hàm chuẩn Random(n) cho giá trị là số nguyên ngẫu nhiên trong khoảng từ 0 đến n - 1.

B. BÀI TẬP ÁP DỤNG

I. Bài tập trắc nghiệm

Hãy chọn phương án trả lời đúng nhất.

Phần cơ bản

11.1. Turbo Pascal có mấy kiểu dữ liệu có cấu trúc?

- | | |
|------------|------------|
| A. 1 loại. | C. 3 loại. |
| B. 2 loại. | D. 4 loại. |

11.2. Mảng là kiểu dữ liệu:

- | | |
|-----------------|--------------|
| A. Phức tạp. | C. Cấu trúc |
| B. Đa cấu trúc. | D. Đơn giản. |

11.3. Số phần tử của mảng là:

- | | |
|-----------------|---------------------|
| A. Vô hạn. | C. Có n phần tử. |
| B. Có giới hạn. | D. Có 1000 phần tử. |

- 11.4.** Kiểu dữ liệu của các phần tử trong mảng là:
- Mỗi phần tử có một kiểu.
 - Kiểu dữ liệu có cấu trúc.
 - Có cùng một kiểu dữ liệu.
 - Có cùng một kiểu đó là kiểu số nguyên.
- 11.5.** Để khai báo một kiểu dữ liệu mảng mới có thể sử dụng:
- Từ khoá khai báo Var.
 - Từ khoá khai báo Type.
 - Từ khoá khai báo Const.
 - Từ khoá khai báo Uses.
- 11.6.** Để khai báo một biến kiểu mảng ta có thể sử dụng:
- Từ khoá Type và từ khoá Var.
 - Từ khoá Var và định nghĩa trực tiếp.
 - Từ khoá Uses và Var.
 - Đáp án A và B đều đúng.
- 11.7.** Cú pháp để khai báo biến mảng một chiều trực tiếp là:
- Var <tên biến mảng> : array[kiểu chỉ số] of <Kiểu phần tử>;
 - Type <tên biến mảng> : array[kiểu chỉ số] of <Kiểu phần tử>;
 - Var <tên biến mảng> : array[kiểu phần tử] of <Kiểu chỉ số>;
 - Var <tên biến mảng> : array[kiểu phần tử] : <Kiểu chỉ số>;
- 11.8.** Cú pháp để khai báo biến mảng một chiều gián tiếp là:
- Type <tên biến mảng> : array[kiểu chỉ số] of <Kiểu phần tử>;
Var : Kiểu biến mảng;
 - Type <tên kiểu mảng> = array[kiểu chỉ số] of <kiểu phần tử>;
Var <tên biến mảng> : <tên kiểu mảng>;
 - Type <tên kiểu mảng> = array[kiểu phần tử] of <kiểu chỉ số>;
Var <tên biến mảng> : <tên kiểu mảng>;
 - Var <tên biến mảng> : <tên kiểu mảng>;
- 11.9.** Mảng một chiều là:
- Mảng có số phần tử tăng dần theo một chiều.
 - Mảng có từ hai kiểu chỉ số trở lên.
 - Mảng chỉ có một kiểu chỉ số.
 - Mảng có duy nhất một phần tử.
- 11.10.** Trong các khai báo dưới đây khai báo nào đúng?
- Type AI = array(1..n) of Integer;
 - Type AA = array[1...n] of Char;
 - Type BB = array[1..100] of Real;
 - Type BB = array[100..1] of Real;
- 11.11.** Trong các khai báo dưới đây khai báo nào đúng?

- A. Arrayreal = array[10..20] of real;
- B. Arrayinteger = array[1..100] of integer;
- C. Logic = array[1..9] of boolean;
- D. Tất cả đều đúng.

11.12. Phát biểu nào dưới đây là đúng:

- A. Tham chiếu tới phần tử của mảng một chiều được xác định bởi tên biến mảng và kiểu phần tử của mảng.
- B. Tham chiếu tới phần tử của mảng một chiều được xác định bởi tên biến mảng và chỉ số phần tử.
- C. Tham chiếu tới phần tử của mảng một chiều được xác định bởi tên biến mảng và chỉ số phần tử được viết trong cặp ngoặc [và].
- D. Tham chiếu tới phần tử của mảng một chiều được xác định bởi tên biến mảng và chỉ số phần tử được viết trong cặp ngoặc "(" và ")".

11.13. Phát biểu nào dưới đây là đúng?

- A. Hai biến mảng có cùng kiểu dữ liệu, cùng số phần tử có thể gán được cho nhau.
- B. Có thể đọc và ghi dữ liệu cho toàn biến mảng mà không phải thực hiện thông qua các phần tử.
- C. Hai biến mảng bất kì có thể gán được cho nhau.
- D. Đọc và ghi biến mảng giống như các loại biến khác.

11.14. Cho đoạn chương trình:

```

Var a,b:array[1..5] of byte;
    l: byte;
Begin
    For i:= 1 to 5 do
        a[i]:= i;
        b:= a;
    End.
```

Khi thực hiện chương trình cho kết quả nào dưới đây?

- A. a = b
- B. b = a
- C. a[1]=b[1]=1; a[2]=b[2]=2; a[3]=b[3]=3; a[4]=b[4]=4; a[5]=b[5]=5;
- D. Cả ba đáp án đều đúng.

11.15. Mảng hai chiều khác mảng một chiều ở chỗ:

- A. Số phần tử khác nhau.
- B. Kiểu của phần tử khác nhau.

C. Giá trị của các phần tử là khác nhau.

D. Mảng một chiều có một chỉ số, mảng hai chiều có hai chỉ số.

11.16. Cú pháp nào dưới đây dùng để khai báo trực tiếp biến mảng hai chiều?

A. Type <tên biến mảng>: array[chi số dòng, chi số cột] of <kiểu phần tử>;

B. Var <tên biến mảng>: array[chi số dòng, chi số cột] of <kiểu phần tử>;

C. Const <tên biến mảng>: array[chi số dòng, chi số cột] of <kiểu phần tử>;

D. Var <tên biến mảng> : <tên kiểu mảng>;

11.17. Cú pháp nào dưới đây dùng để khai báo gián tiếp biến mảng hai chiều?

A. Type <tên biến mảng> = array[Kiểu chỉ số] of <kiểu phần tử>;

Var <tên biến mảng> : <tên kiểu mảng>;

B. Type <tên kiểu mảng> = array[Chỉ số dòng, chỉ số cột] of <kiểu phần tử>;

C. Type <tên kiểu mảng> = array[Chỉ số dòng, chỉ số cột] of <kiểu phần tử>;

Var <tên biến mảng> : <tên kiểu mảng>;

D. Type <tên biến mảng> = array[Chỉ số dòng, chỉ số cột] of <kiểu phần tử>;

Var <tên biến mảng> : <tên kiểu mảng>;

11.18. Trong các khai báo dưới đây khai báo nào sai?

A. Var M: array[1..5, 1..10] of Integer;

B. Var N: array[1..3] of array[1..5] of Real;

C. Type P: array[1..50, 1..10] of Char;

D. Var N: array[5..3] of array[7..5] of Real;

11.19. Giả sử đã khai báo biến mảng và các biến khác, đoạn chương trình dưới đây làm gì?

For i := 1 to n do

Begin

Write('a[', i, '=');

Readln(a[i]);

End;

A. Đếm số phần tử của mảng A.

B. Nhập giá trị cho các phần tử của mảng A.

C. Viết ra giá trị a[i].

D. Hiện thị các phần tử của mảng A.

11.20. Trong ngôn ngữ lập trình Pascal, với khai báo mảng $A: \text{Array}[1..100, 1..100] \text{ of integer}$; thì việc truy xuất đến các phần tử được thực hiện như thế nào?

- A. $A[i], [j]$ B. $A[i][j]$ C. $A[i;j]$ D. $A[i,j]$

11.21. Trong ngôn ngữ lập trình Pascal, với khai báo mảng $A: \text{Array}[1..100] \text{ of integer}$; thì việc truy xuất đến các phần tử như sau:

- A. (A_i, j) B. $A[i][j]$ C. $A[i;j]$ D. $A[i,j]$

11.22. Trong ngôn ngữ lập trình Pascal, trong quá trình nhập dữ liệu của mảng một chiều A, để các phần tử hiển thị như trong cửa sổ chương trình ta viết lệnh như sau:

- A. `Write('A['i,']='); Readln(A[i]);`
 B. `Write('A[i]='); Readln(A[i]);`
 C. `Write('A['i']='); Readln(A[i]);`
 D. `Write("A["i,"]=""); Readln(A[i]);`

11.23. Trong ngôn ngữ lập trình Pascal, trong quá trình nhập dữ liệu của mảng hai chiều A, để các phần tử hiển thị như trong cửa sổ chương trình ta viết lệnh như sau:

- A. `Write('A['i,j,']='); Readln(A[i,j]);`
 B. `Write('A['i,','j,']='); Readln(A[i,j]);`
 C. `Write('A['i,j']='); Readln(A[i,j]);`
 D. `Write("A["i,j,"]=""); Readln(A[i,j]);`

11.24. Trong ngôn ngữ lập trình Pascal, trong quá trình xuất dữ liệu của mảng hai chiều A có m dòng và n cột, để các phần tử hiển thị đúng như mô hình của mảng hai chiều ta viết lệnh như sau:

- A. `For i:=1 to m do`
 `For j:=1 to n do`
 `Write (A[i,j]:5);`

- C. `For i:=1 to m do`
 `Begin`
 `For j:=1 to n do`
 `Writeln;`
 `End;`

- B. `For i:=1 to m do`
 `Begin`
 `For j:=1 to n do`
 `Write (A[i,j]:5);`
 `End;`

- D. `For i:= 1 to m do`
 `Begin`
 `For j:=1 to n do`
 `Write (A[i,j]:5);`
 `Writeln;`
 `End;`

11.25. Trong ngôn ngữ lập trình Pascal, để đổi giá trị hai phần tử mảng một chiều A tại hai vị trí i và j, ta viết mã lệnh như sau:

A. $A[j] := \text{Tag};$
 $A[i] := A[j];$
 $A[j] := \text{Tag};$
 B. $\text{Tag} := A[i];$
 $A[i] := A[j];$
 $A[j] := \text{Tag};$

C. $\text{Tag} := A[i];$
 $A[j] := A[i];$
 $A[j] := \text{Tag};$
 D. $\text{Tag} := A[i];$
 $A[i] := A[j];$
 $\text{Tag} := A[j];$

11.26. Trong ngôn ngữ lập trình Pascal, đoạn chương trình sau thực hiện việc nào trong các việc sau (A là mảng số có N phần tử)?

$S := 0;$

For $i := 1$ to N Do $S := S + A[i];$

- A. Tính tổng các phần tử của mảng A
- B. In ra màn hình mảng A
- C. Đếm số phần tử của mảng A
- D. Không thực hiện việc nào trong ba việc trên

11.27. Trong ngôn ngữ lập trình Pascal, với khai báo như sau:

Type mang = array[1..100] of integer;

Var a, b: mang;

c: array[1..100] of integer;

Câu lệnh nào dưới đây là hợp lệ?

- A. $a := b;$
- B. $b := c;$
- C. $c := b;$
- D. $a := c;$

Phần nâng cao.

11.28. Giả sử đã khai báo biến mảng và các biến khác, đoạn chương trình dưới đây làm gì?

For $i := 1$ to m do

Begin

Writeln('a[', i , ']=', $a[i]$);

End;

- A. Hiện thị tất cả các giá trị trên một dòng.
- B. In ra chỉ số các phần tử của mảng.
- C. Hiện thị giá trị các phần tử của mảng, mỗi phần tử nằm trên một dòng.
- D. Viết ra màn hình câu $a[i]$

11.29. Giả sử đã khai báo biến mảng và các biến khác, đoạn chương trình dưới đây làm gì?

.....


```

For i := 1 to n-1 do
  For j := i+1 to n do
    If a[i] > a[j] then
      Begin
        Tg:= a[i];
        a[i]:= a[j];
        a[j]:=tg;
      End;

```

.....

- A. Hoán đổi các giá trị của mảng cho nhau.
- B. Sắp xếp các phần tử của mảng theo trật tự giảm dần.
- C. Sắp xếp các phần tử của mảng theo trật tự tăng dần.
- D. Sắp xếp lại giá trị chỉ số của mảng.

11.30. Giả sử đã khai báo biến mảng và các biến khác, đoạn chương trình dưới đây làm gì?

```

For i := 1 to n do
  For j:= 1 to m do
    Begin
      Write('a[', i, j, ']=');
      Readln(a[i,j]);
    End;

```

- A. Đếm số phần tử của mảng hai chiều A.
- B. Nhập các phần tử cho mảng hai chiều A.
- C. Sắp xếp các phần tử của mảng hai chiều A.
- D. Hiện thị các phần tử của mảng hai chiều A.

11.31. Giả sử đã khai báo biến mảng và các biến khác, đoạn chương trình dưới đây làm gì?

```

For i := 1 to n do
  For j:= 1 to m do
    Begin
      Writeln('a[', i, j, ']=', a[i,j]);
    End;

```

- A. Tính tổng các phần tử của mảng hai chiều A.
- B. In ra chỉ số các phần tử của mảng hai chiều A.
- C. Hiện thị giá trị các phần tử của mảng hai chiều A.
- D. Tìm kiếm một phần tử trong mảng hai chiều A.

11.32. Cho đoạn chương trình:

```
Var a:array[1..10] of Integer;  
    l: byte;  
Begin  
    For i:= 1 to 10 do  
        a[i]:= i;  
    For i:= 1 to 10 do  
        If a[i] mod 2 = 0 then Write(a[i]:2);  
    End.
```

Khi thực hiện chương trình cho kết quả nào dưới đây?

- A. 1 2 3 4 5 6 7 8 9 10
- B. 1 3 5 7 9
- C. 2 4 6 8 10
- D. 10 8 6 4 2 0

11.33. Cho đoạn chương trình:

```
Var M:array[1..10] of Integer;  
    l: byte;  
Begin  
    For i:= 1 to 10 do  
        M[i]:= i;  
    For i:= 1 to 10 do  
        Write(m[5]:1, m[7]:1, m[9]:1);  
    End.
```

Khi thực hiện chương trình cho kết quả nào dưới đây?

- | | |
|----------|----------|
| A. 2 4 6 | C. 3 7 9 |
| B. 1 3 5 | D. 5 7 9 |

11.34. Cho đoạn chương trình:

```
Var B:array[1..100] of Integer;  
    i: Integer;  
Begin  
    For i:= 1 to 100 do  
        B[i]:= i;  
    For i:= 1 to 100 do  
        If B[i] = 50 then Write(B[i]/10:2:0, i);  
        Readln;  
    End.
```

Khi thực hiện chương trình cho kết quả nào dưới đây?

- A. 50 50
- B. 5 i
- C. 5 50
- D. 50 5

11.35. Cho đoạn chương trình:

```

Var Q : array[1..10] of Integer;
    i: Integer; dem: integer;
Begin
    Q[1]:= 1; Q[2]:= 1; Q[3]:= 5; Q[4]:= 5; Q[5]:= 5;
    Q[6]:= 8; Q[7]:= 9; Q[8]:= 0; Q[9]:= 5; Q[10]:= 7;
    Dem:=0;
    For i:= 1 to 10 do
        If Q[i] = 5 then dem:=dem+1;
        Writeln(dem);
        Readln;
    End.

```

Khi thực hiện chương trình cho kết quả nào dưới đây?

- A. 2.
- B. 3.
- C. 4.
- D. 5.

11.36. Cho đoạn chương trình:

```

Var Q : array[1..10] of Integer;
    i,j,tg: Integer; dem: integer;
Begin
    Q[1]:= 11; Q[2]:= 33; Q[3]:= 5; Q[4]:= 6; Q[5]:= 15;
    Q[6]:= 8; Q[7]:= 19; Q[8]:= 0; Q[9]:= 25; Q[10]:= 7;
    For i:= 1 to 9 do
        For j:= i+1 to 10 do
            If Q[i] > Q[j] then
                Begin tg:= Q[i];
                    Q[i]:=Q[j];
                    Q[j]:= tg;
                End;
        For i:= 1 to 10 do
            Write(Q[i]:2);
            Readln;
        End.

```

Khi thực hiện chương trình cho kết quả nào dưới đây?

- A. 11 33 5 6 15 8 19 0 25 7
- B. 0 5 6 7 8 11 15 19 25 33
- C. 0 5 6 7 19 25 33 8 11 15
- D. 33 25 19 15 11 8 7 6 5 0

11.37. Cho đoạn chương trình:

```

Var B:array[1..10] of Integer;
    l, tong: Integer;
Begin
    For i:= 1 to 10 do
        B[i]:=i;
        Tong:=0;
    For i:= 1 to 10 do
        Tong:= tong+B[i];
    Write(tong);
    Readln;

```

End.

Khi thực hiện chương trình cho kết quả nào dưới đây?

- A. 40
- B. 45
- C. 50
- D. 55

11.38. Cho đoạn chương trình:

```

Var S:array[1..10] of Integer;
    l, tong: Integer;
Begin
    For i:= 1 to 10 do
        S[i]:= i;
        Tong:=0;
    For i:= 1 to 10 do
        If s[i] mod 2 = 0 then
            Tong:= tong+S[i];
    Write(tong);
    Readln;

```

End.

Khi thực hiện chương trình cho kết quả nào dưới đây?

- A. 30
- B. 40
- C. 50
- D. 60

II. Bài tập tự luận

Phần cơ bản.

- 11.1.** Viết chương trình nhập vào một mảng các số nguyên rồi hiển thị các số đó ra màn hình.
- 11.2.** Viết chương trình nhập vào một mảng các số nguyên rồi thực hiện các công việc sau đây:
 - a. Tính tổng các phần tử của mảng.
 - b. Đếm số các phần tử có giá trị dương và tính tổng của chúng.
 - c. Đếm số các phần tử có giá trị âm và tính tổng của chúng.
 - d. Tính trung bình cộng các giá trị trong mảng.
- 11.3.** Viết chương trình nhập vào một mảng các số thực rồi thực hiện các công việc sau đây:
 - a. Tìm số dương đầu tiên và chỉ số của nó trong mảng.
 - b. Tìm số âm đầu tiên và chỉ số của nó trong mảng.
 - c. Tìm số dương cuối cùng và chỉ số của nó trong mảng.
 - d. Tìm số âm cuối cùng và chỉ số của nó trong mảng.
- 11.4.** Viết chương trình nhập vào một mảng các số nguyên rồi thực hiện các công việc sau đây:
 - a. Tìm giá trị lớn nhất và in ra chỉ số của nó.
 - b. Tìm giá trị bé nhất và in ra chỉ số của nó.
 - c. Tìm giá trị âm lớn nhất và in ra chỉ số của nó.
 - d. Tìm giá trị dương lớn nhất và in ra chỉ số của nó.
- 11.5.** Viết chương trình nhập vào một mảng một chiều chứa các số nguyên sau đó sắp xếp các giá trị của mảng theo trật tự tăng dần rồi in ra màn hình.
- 11.6.** Viết chương trình nhập vào một mảng một chiều chứa các số nguyên sau đó sắp xếp các giá trị của mảng theo trật tự giảm dần rồi in ra màn hình.
- 11.7.** Viết chương trình nhập vào một mảng một chiều chứa các số nguyên sau đó thực hiện:
 - a. In ra danh sách các số chẵn và tổng của chúng.
 - b. In ra danh sách các số lẻ và tổng của chúng.
- 11.8.** Viết chương trình nhập vào một mảng một chiều chứa các số thực sau đó thực hiện kiểm tra xem trong mảng có bao nhiêu phần tử có giá trị bằng một giá trị cho trước và nhập từ bàn phím.

11.9. Viết chương trình nhập vào một dãy n số nguyên và thực hiện dịch chuyển k phần tử đầu dãy về cuối dãy ($k < n$). Ví dụ $n=5$, $k=3$ và ta nhập vào dãy 3, 5, 7, 1, 2 thì dãy kết quả là 1, 2, 3, 5, 7.

11.10. Viết chương trình tìm phần tử thứ n của dãy FIBONACCI.

Phần nâng cao

11.11. Viết chương trình nhập một dãy số đôi một khác nhau và:

- Cho biết độ dài lớn nhất của dãy gồm các số 0 liên tiếp.
- Tìm dãy con liên tiếp tăng và có tổng lớn nhất.

11.12. Viết chương trình:

- Bổ sung một phần tử vào mảng một chiều.
- Loại bỏ một phần tử ra khỏi mảng một chiều.

11.13. Viết chương trình nhập vào một dãy số và cho biết dãy đó có bao nhiêu phần tử đôi một khác nhau đồng thời thông báo các phần tử đó lên màn hình.

11.14. Viết chương trình tìm cách lấy ra n nghìn đồng từ các loại tiền 1 nghìn, 2 nghìn, 5 nghìn, 10 nghìn, 20 nghìn, 50 nghìn, 100 nghìn sao cho tổng số tờ là bé nhất. (Số tiền n , số tờ của mỗi loại tiền được nhập từ bàn phím).

11.15. Viết chương trình nhập vào một ma trận vuông cấp n rồi cho biết:

- Tổng tất cả các phần tử nằm trên đường chéo chính.
- Tổng tất cả các phần tử nằm phía trên đường chéo chính.

11.16. Viết chương trình nhập vào hai ma trận vuông cấp n (với các phần tử nguyên) rồi tính tổng và tích của hai ma trận đó.

11.17. Viết chương trình nhập vào một ma trận cấp $m \times n$ rồi tính tổng tất cả các phần tử thoả mãn $i - j = k$ (m, n, i, j, k nhập từ bàn phím, $0 < i, j, k < n$).

11.18. Viết chương trình xổ số với cơ cấu như sau:

- Ba giải tư, quay hai số.
- Hai giải ba, quay ba số.
- Một giải nhì, quay bốn số.
- Một giải nhất, quay năm số.
- Một giải đặc biệt, quay sáu số.

11.19. Viết chương trình nhập vào từ bàn phím một dãy số có 10 phần tử và một số x . Tìm trong dãy có hai phần tử liên tiếp nào mà tổng của chúng bằng x không?

11.20. Viết chương trình nhập vào mảng hai chiều a có m dòng và n cột sau đó thực hiện:

- Tìm giá trị lớn nhất và nhỏ nhất trên mỗi hàng.
- Tìm giá trị lớn nhất và nhỏ nhất trên mỗi cột.
- Tìm giá trị lớn nhất và nhỏ nhất của mảng.
- Tìm trong mảng có bao nhiêu phần tử bằng phần tử lớn nhất.

Bài 12

Kiểu Xâu

1. PHƯƠNG PHÁP GIẢI TOÁN

1. *Khái niệm xâu, khai báo, tham chiếu*

- Xâu kí tự là dãy các kí tự trong bảng mã ASCII, mỗi kí tự được gọi là một phần tử của xâu. Số lượng các kí tự trong xâu được gọi là độ dài xâu. Xâu có độ dài bằng không là xâu rỗng.

- Khai báo biến: `VAR tên_biến : STRING[độ dài xâu];`

- Tham chiếu đến từng kí tự trong xâu: `tên_biến[chỉ_số]`

2. *Phép ghép xâu*

- Phép ghép xâu: kí hiệu là `+`, được sử dụng để ghép nhiều xâu thành một xâu.

- Các phép so sánh: `=`, `<>`, `>`, `<`, `<=`, `>=` thực hiện việc so sánh hai xâu. Xâu A được xem là lớn hơn xâu B nếu như kí tự khác nhau đầu tiên giữa chúng kể từ trái sang trong xâu A có chỉ số trong bảng mã ASCII là lớn hơn. Nếu A và B là các xâu có độ dài khác nhau và A là đoạn đầu của B thì A nhỏ hơn B.

3. *Một số thủ tục và hàm chuẩn.*

- Thủ tục `Delete(st,vt,n)` thực hiện việc xoá đi trong xâu st gồm n kí tự, bắt đầu từ vị trí vt.

- Thủ tục `Insert(st1,st2,vt)` thực hiện việc chèn xâu st1 vào xâu st2 bắt đầu từ vị trí vt.

- Thủ tục `Str(value, st)` thực hiện việc đổi giá trị số value thành xâu và gán cho biến xâu st.

Thủ tục `Val(st, var, code)` thực hiện việc đổi xâu st biểu diễn số và gán cho biến var, nếu việc đổi thành công thì biến code có giá trị bằng 0, ngược lại nếu xâu st biểu diễn số bị lỗi thì khi đó var không xác định còn code có giá trị là vị trí lỗi trong biểu diễn số của xâu st.

- Hàm Copy(st,vt,n) cho giá trị là một xâu kí tự được lấy trong xâu st, gồm n kí tự liên tiếp bắt đầu từ vị trí vt của xâu st.

- Hàm Length(st) cho giá trị là số lượng kí tự của xâu st.

- Hàm Pos(st1,st2) cho giá trị là vị trí xuất hiện đầu tiên của xâu st1 trong xâu st2.

- Hàm UpCase(ch) cho giá trị là kí tự hoa tương ứng với kí tự ch.

- Hàm Concat(st1, st2, ..stn) thực hiện ghép nối các chuỗi với nhau theo thứ tự để được một chuỗi có chiều dài là tổng chiều dài các chuỗi.

Một số ví dụ:

Sử dụng thủ tục Delete:

Giá trị của St	Thao tác	Kết quả
'ABCDEF'	Delete(St, 5, 2);	'ABCD'
'BONG HONG'	Delete(St, 1, 5);	'HONG'

Sử dụng thủ tục Insert:

Giá trị S1	Giá trị S2	Thao tác	Kết quả
'PC'	'IBM586'	Insert (S1, S2, 4);	'IBM PC 586'
'M'	'Hình.1'	Insert (S1, S2, 6);	'Hình. M1'

Sử dụng thủ tục Copy:

Giá trị S	Biểu thức	Kết quả
'BAI HOC THU 9'	Copy(S, 9, 5);	'THU 9'

Sử dụng hàm length:

Giá trị S	Biểu thức	Kết quả
'500 kí tu'	Length(S)	9

Sử dụng hàm Pos:

Giá trị S	Biểu thức	Kết quả
'abcdef'	Pos('cd',S2)	3
'abcdef'	Pos('k',S2)	0

Sử dụng hàm Upcase:

Giá trị Ch	Biểu thức	Kết quả
'd'	UpCase (ch)	'D'
'E'	UpCase (ch)	'E'

B. BÀI TẬP ÁP DỤNG

I. Bài tập trắc nghiệm

Hãy chọn phương án trả lời đúng nhất.

Phần cơ bản

12.1. Xâu là kiểu dữ liệu:

- A. Liệt kê.
- B. Có cấu trúc.
- C. Đơn giản chuẩn.
- D. Vô hướng

12.2. Xâu là kiểu dữ liệu có độ dài tối đa là:

- A. 120 kí tự.
- B. 256 kí tự.
- C. 255 kí tự.
- D. 128 kí tự.

12.3. Để định nghĩa một kiểu xâu có độ dài xác định sử dụng cú pháp nào?

- A. Type <tên kiểu xâu>: string;
- B. Type <tên kiểu xâu> = string;
- C. Type <tên kiểu xâu> = string[n];
- D. Type <tên kiểu xâu> := string[n];

12.4. Để khai báo biến kiểu xâu gián tiếp ta sử dụng cú pháp nào?

- A. Type <tên kiểu xâu> = string[n];
Var <tên biến>: <tên kiểu xâu>;
- B. Type <tên kiểu xâu>: string[n];
Var <tên biến>: <tên kiểu xâu>;
- C. Type <tên kiểu xâu> = string;
Var <tên biến>: <tên kiểu xâu>;
- D. Var <tên biến>: <tên kiểu xâu>;

12.5. Để khai báo biến kiểu xâu trực tiếp ta sử dụng cú pháp nào?

- A. Var <tên biến>: <tên kiểu>;
- B. Var <tên biến>: string[n];
- C. Var <tên biến> = String[n];
- D. Var <tên biến> = String;

12.6. Trong các khai báo sau khai báo nào sai?

- A. type st30 = string[30];
- B. Var hoten: string[30];
- C. Type st10 = string[10];
Var s1: st10;
- D. Var s2: string[20];

12.7. Trong các phát biểu sau phát biểu nào sai:

- A. Xâu không có kí tự nào được gọi là xâu rỗng.
- B. Có thể sử dụng các lệnh đọc và ghi đối với kiểu dữ liệu xâu.
- C. Xâu có chiều dài xác định.
- D. Trong xâu, byte cuối cùng chứa độ dài của xâu.

12.8. Trên kiểu dữ liệu xâu có các phép toán nào?

- A. Phép so sánh và phép gán
- B. Phép so sánh và phép nối
- C. Phép gán và phép nối.
- D. Phép gán, phép nối và phép so sánh.

12.9. Trong các phát biểu sau, phát biểu nào đúng?

- A. Hai xâu bằng nhau là hai xâu có cùng số phần tử.
- B. Hai xâu khác nhau khi số phần tử của chúng khác nhau.
- C. Hai xâu có độ dài bằng nhau khi tất cả các cặp kí tự đều giống nhau.
- D. Hai xâu có cùng độ dài luôn bằng nhau.

12.10. Trong các phát biểu sau, phát biểu nào sai?

- A. Khi so sánh hai xâu từ trái qua phải theo từng cặp kí tự thì xâu nào có kí tự lớn hơn thì xâu đó lớn hơn.
- B. Khi so sánh hai xâu từ trái qua phải theo từng cặp kí tự thì xâu nào có kí tự nhỏ hơn thì xâu đó nhỏ hơn.
- C. Hai xâu khác nhau khi có ít nhất một cặp kí tự khác nhau.
- D. Mọi xâu chứa toàn kí tự trắng đều bằng nhau.

12.11. Cho `st1 := 'abcd'; st2 := 'ABC';` khi đó `st1+st2` cho kết quả nào?

- A. `abcd`.
- B. `aabbccdd`.
- C. `abcdABC`.
- D. `ABCabcd`.

12.12. Trong các phép toán sau phép nào trả về giá trị đúng?

- A. `'abcdh' > 'abcdef'`.
- B. `'abc' = 'ABC'`.
- C. `'cba' < 'abc'`.
- D. `'ccb' < 'abcdef'`.

12.13. Phép so sánh chuỗi `s1 = 'Hoa học tro'` và chuỗi `s2 = 'Hoa học Tro'` nhận kết quả nào?

- A. `s1 = s2`
- B. `s1 > s2`
- C. `s1 < s2`
- D. `s1 < > s2`.

12.14. Cho chuỗi `st`, cho `Num` và `pos` là 2 biến kiểu nguyên chỉ số kí tự và vị trí bất kì trong xâu. Thủ tục `Delete(st, pos, num)` có chức năng?

- A. Xóa một chuỗi có độ dài bất kì.
- B. Xóa kí tự trong chuỗi `st`.
- C. Xóa trong xâu `st` `num` kí tự bắt đầu từ vị trí `pos`.
- D. Xóa trong xâu `st` `pos` kí tự bắt đầu từ vị trí `num`.

12.15. Cho chuỗi `st1, st2`. Cho `pos` là biến kiểu nguyên chỉ vị trí bất kì trong xâu. Thủ tục `Insert(st2, st1, pos)` có chức năng?

- A. Chèn xâu `st2` vào xâu `st1` bắt đầu ở vị trí `pos`.

- B. Chèn xâu st1 vào xâu st2 bắt đầu ở vị trí pos.
- C. Chèn xâu st2 vào xâu st1 bắt đầu ở vị trí cuối xâu.
- D. Chèn xâu st1 vào xâu st2 bắt đầu ở vị trí đầu xâu.

Phân nâng cao

12.16 Cho thủ tục Str(value,st), trong đó value là một biểu thức số học có thể ghi dưới dạng quy cách, st là một chuỗi. Thủ tục có chức năng?

- A. Đổi giá trị xâu st thành số và đưa vào giá trị value.
- B. Đổi giá trị của số value thành xâu và gán cho biến st.
- C. Đổi giá trị xâu st thành số.
- D. Đổi giá trị của số value thành xâu.

12.17 Cho thủ tục Val(st, var, code), trong đó var là một biến kiểu nguyên hay thực, code là biến kiểu nguyên, st là một chuỗi. Thủ tục có chức năng?

- A. Thủ tục này dùng để đổi xâu thành số và đưa kết quả vào code, nếu việc đổi thành công thì biến code có giá trị bằng 0, ngược lại nếu xâu st biểu diễn số bị lỗi thì khi đó var không xác định còn code có giá trị là vị trí lỗi trong biểu diễn số của xâu st.
- B. Thủ tục dùng để đổi số thành xâu và đưa kết quả vào var, nếu việc đổi thành công thì biến code có giá trị bằng 0, ngược lại nếu xâu st biểu diễn số bị lỗi thì khi đó var không xác định còn code có giá trị là vị trí lỗi trong biểu diễn số của xâu st.
- C. Thủ tục này sẽ đổi xâu st biểu diễn số thành số và gán cho biến var, nếu việc đổi thành công thì biến code có giá trị bằng 0, ngược lại nếu xâu st biểu diễn số bị lỗi thì khi đó var không xác định còn code có giá trị là vị trí lỗi trong biểu diễn số của xâu st.
- D. Thủ tục này sẽ đổi xâu st biểu diễn số thành số và gán cho biến var, nếu việc đổi thành công thì biến code có giá trị bằng 1, ngược lại nếu xâu st biểu diễn số bị lỗi thì khi đó var không xác định còn code có giá trị là vị trí lỗi trong biểu diễn số của xâu st.

12.18. Cho hàm length(st) trong đó st là biến kiểu xâu. Chức năng của hàm là:

- A. Hàm trả về chiều dài các kí tự trống.
- B. Hàm trả về độ dài thực của xâu st.
- C. Hàm trả về độ dài tối đa của xâu.
- D. Hàm trả về độ dài xâu st sau khi đã bỏ kí tự trắng.

12.19. Cho hàm Copy(st, pos, num), trong đó num và pos là 2 biến kiểu nguyên chỉ số kí tự và vị trí bất kì trong chuỗi, st là một chuỗi. Chức năng của hàm:

- A. Cho kết quả là xâu con của xâu st gồm num kí tự bắt đầu từ vị trí pos.
- B. Cho kết quả là xâu con của xâu st gồm num kí tự.
- C. Sao chép xâu st thành một xâu khác.
- D. Sao chép num kí tự vào xâu st.

12.20. Cho hàm concat(st1, st2, st3,...,stn), $n > 2$, st1, ..stn là các xâu. Hàm trả về kết quả:

- A. Cho xâu tổng bằng cách nối các xâu lại với nhau.
- B. Cho kết quả là xâu được ghép nối theo thứ tự các xâu $st1, st2, st3, \dots, stn$.
- C. Trả về độ dài của xâu tổng sau khi đã nối các xâu theo thứ tự các xâu $st1, st2, st3, \dots, stn$.
- D. Cho kết quả là xâu được ghép nối theo thứ tự các xâu $stn, \dots, st3, st2, st1$.

12.21. Cho hàm $\text{Pos}(\text{st1}, \text{st2})$, trong đó st1 , st2 là các xâu. Hàm trả về kết quả:

- A. Vị trí xuất hiện của xâu st1 trong xâu st2.
B. Vị trí xuất hiện của xâu st2 trong xâu st1.
C. Vị trí xuất hiện đầu tiên của xâu st1 trong xâu st2. Nếu xâu st1 không có mặt trong xâu st2 thì hàm trả về kết quả là 0.
D. Vị trí xuất hiện đầu tiên của xâu st1 trong xâu st2. Nếu xâu st1 không có mặt trong xâu st2 thì hàm báo lỗi.

12.22. Cho $st = \text{'abcde'}$. Kết quả khi gọi thủ tục $\text{Delete}(st, 2, 3)$ là:

- A. st = 'ab' C. st = 'ac'
- B. st = 'bc' D. st = 'ae'

12.23. Cho `st1 = 'abc'`; `st2 = 'def'`. Kết quả gọi `Insert(st2,st1,3)` là:

- A. 'abcdef't
B. 'abdefct'
C. 'abcfted'
D. 'eftcabd'

12.24. Kết quả thực hiện của thủ tục str(123.456,st) là:

- A. st = 123.456
B. st = 123
C. st = '123.456'
D. st = '123456'

12.25. Cho $st = '123.456'$, khi thực hiện thủ tục $Val(st, a, k)$ cho kết quả:

- A. `a = 123.456` C. `a = '123.456'`
B. `a = 123456` D. `a = '123456'`

- 12.26.** Cho `st = '12q.456'`, khi thực hiện thủ tục `Val(st, a, k)` cho kết quả:
- A. `a = 12.456`
 - B. `a = 12`
 - C. `a = 0` và `k = 3`
 - D. `a` không có giá trị xác định và `k = 0`
- 12.27.** Cho `st = 'chuc cac ban may man'`, kết quả hàm `length(st)` là:
- A. 16.
 - B. 18.
 - C. 20.
 - D. 22.
- 12.28.** Kết quả của việc thực hiện hàm `copy('abcdef', 3, 2)` là:
- A. `'ab'`
 - B. `'cd'`
 - C. `'ef'`
 - D. `'df'`
- 12.29.** Kết quả của việc thực hiện hàm `concat('abc', 'de', '123')` cho kết quả là:
- A. `'abcde'`
 - B. `'123abcde'`
 - C. `'abcde123'`
 - D. `'abc123'`
- 12.30.** Kết quả của việc thực hiện hàm `Pos('ab', 'deabcab')` là:
- A. 2.
 - B. 3.
 - C. 4.
 - D. 5.
- 12.31.** Kết quả của việc thực hiện hàm `Pos('ab', 'defbcad')` là:
- A. 0.
 - B. 1.
 - C. 2.
 - D. 3.
- 12.32.** Trong ngôn ngữ lập trình Pascal, xâu kí tự là:
- A. Mảng các kí tự;
 - B. Dãy các kí tự trong bảng mã ASCII;
 - C. Tập hợp các chữ cái trong bảng chữ cái tiếng Anh;
 - D. Tập hợp các chữ cái và chữ số trong bảng chữ cái tiếng Anh.
- 12.33.** Trong ngôn ngữ lập trình Pascal, cách khai báo xâu kí tự nào sau đây là đúng?
- A. `S : File of String;`
 - B. `S : File of char;`
 - C. `S : String;`
 - D. Cả 3 câu đều đúng.
- 12.34.** Trong ngôn ngữ lập trình Pascal, xâu kí tự không có kí tự nào gọi là?
- A. Xâu không;
 - B. Xâu rỗng;
 - C. Xâu trắng;
 - D. Không phải là xâu kí tự.
- 12.35.** Trong ngôn ngữ lập trình Pascal, phần tử đầu tiên của xâu kí tự mang chỉ số là?
- A. 0
 - B. Do người lập trình khai báo.
 - C. 1
 - D. Không có chỉ số.

12.36. Trong ngôn ngữ lập trình Pascal, khai báo nào trong các khai báo sau là sai khi khai báo chuỗi ký tự?

A. S : String;

C. S : String[256];

B. X1: String[100];

D. X1: String[1].

12.37. Trong ngôn ngữ lập trình Pascal, sau khi chương trình thực hiện xong đoạn chương trình sau, giá trị của biến S là?

S:= 'Ha Noi mua thu';

Delete(S, 7, 8);

Insert('Mua thu', S, 1);

A. Ha Noi mua thu

C. Mua thu Ha Noi

B. Mua thu Ha Noi mua thu

D. Ha Noi

12.38. Trong ngôn ngữ lập trình Pascal, đoạn chương trình sau thực hiện việc gì?

If (ch >= 'A') and (ch <= 'Z') Then ch := Chr(Ord(ch)+32);

A. Xoá ký tự có trong biến ch

B. Biến ch thành chữ thường nếu ch là chữ hoa

C. Không thực hiện gì

D. Biến ch thành chữ hoa

12.39. Trong ngôn ngữ lập trình Pascal, thủ tục chèn chuỗi S1 vào chuỗi S2 bắt đầu từ vị trí vt được viết.

A. Insert(vt, S1, S2)

C. Insert(S1, vt, S2)

B. Insert(S1, S2, vt)

D. Insert(S2, S1, vt)

12.40. Trong ngôn ngữ lập trình Pascal, với chuỗi ký tự có các phép toán là:

A. Phép cộng, trừ, nhân, chia

B. Phép cộng và phép trừ

C. Chỉ có phép cộng

D. Phép ghép chuỗi và phép so sánh

12.41. Trong ngôn ngữ lập trình Pascal, với chuỗi ký tự ta có thể:

A. So sánh hai chuỗi ký tự

B. Gán biến chuỗi cho biến chuỗi

C. Gán một ký tự cho biến chuỗi

D. Cả ba việc này.

12.42. Trong ngôn ngữ lập trình Pascal, hai chuỗi ký tự được so sánh dựa trên:

A. Mã của từng ký tự trong các chuỗi lần lượt từ trái sang phải

B. Độ dài tối đa của hai chuỗi

C. Độ dài thực sự của hai chuỗi

D. Số lượng các ký tự khác nhau trong chuỗi

12.43. Trong ngôn ngữ lập trình Pascal, hàm Upcase(Ch) cho kết quả là:

- A Chữ cái in hoa tương ứng với Ch
- B Xâu Ch gồm toàn chữ hoa
- C Xâu Ch toàn chữ thường
- D Biến Ch thành chữ thường

12.44. Trong ngôn ngữ lập trình Pascal, đoạn chương trình sau thực hiện công việc gì? (giá trị của biến dem)

dem := 0;

For ch := 'a' To 'z' Do

 If Pos(ch, S) <> 0 Then dem := dem + 1

- A Đếm số lượng kí tự khác dấu cách của xâu S
- B Đếm số lượng kí tự là chữ cái in hoa của xâu S
- C Đếm số lượng kí tự là chữ cái thường trong xâu S
- D Đếm số lượng là chữ cái thường khác nhau có trong xâu S

12.45. Trong ngôn ngữ lập trình Pascal, đoạn chương trình sau thực hiện công việc gì?

X := Length(S);

For i := X downto 1 Do

 If S[i] = " " Then Delete(S,i,1); {là một dấu cách}

- A. Xoá dấu cách đầu tiên trong xâu kí tự S
- B. Xoá dấu cách thừa trong xâu kí tự S
- C. Xoá dấu cách tại vị trí cuối cùng của xâu S
- D. Xoá mọi dấu cách của xâu S

12.46. Trong ngôn ngữ lập trình Pascal, đoạn chương trình sau thực hiện công việc gì?

S1 := 'anh'; S2 := 'em'; i := POS(S2, S1);

While i <> 0 Do

Begin

 Delete(S,i,2);

 Insert(S1,S,i);

 i := POS(S2,S);

End;

- A. Thay toàn bộ cụm từ "anh" trong xâu S bằng cụm từ 'em'.
- B. Thay toàn bộ cụm từ "em" trong xâu S bằng cụm từ "anh".
- C. Thay cụm từ "em" đầu tiên trong xâu S bằng cụm từ "anh".
- D. Không làm gì cả.

12.47. Trong ngôn ngữ lập trình Pascal, để xoá đi kí tự đầu tiên của xâu kí tự S ta viết:

- A. Delete(S, 1, 1);
- B. Delete(S, i, 1); {i là biến có giá trị bất kì};
- C. Delete(S, Length(S), 1);
- D. Delete(S, 1, i); {i là biến có giá trị bất kì}.

12.48. Trong ngôn ngữ lập trình Pascal, để tìm vị trí xuất hiện đầu tiên của xâu "hoa" trong xâu S ta có thể viết bằng cách nào trong các cách sau?

- A. S1:= 'hoa'; I := Pos(S1, 'hoa');
- B. I := Pos('hoa', S);
- C. I := Pos(S, 'hoa');
- D. I := Pos('hoa', 'hoa');

12.49. Trong ngôn ngữ lập trình Pascal, để in một xâu kí tự ra màn hình theo thứ tự ngược lại của các kí tự trong xâu (vd: abcd thì in ra dcba), đoạn chương trình nào sau đây thực hiện việc này?

- A. For i := 1 To Length(S) Do Write(S[i]);
- B. For i := Length(S) Downto 1 Do Write(S[i]);
- C. For i := Length(S) Downto 1 Do Write(S);
- D. For i := 1 To Length(S) Div 2 Do Write(S[i]);

12.50. Trong ngôn ngữ lập trình Pascal, để đếm số kí tự là kí tự số trong xâu S, đoạn chương trình nào trong các đoạn sau thực hiện việc này (biến d dùng để đếm).

- A. d := 0;
For i := 1 To length(S) Do
If (S[i] >= 0) and (S[i] <= 9) Then d := d + 1;
- B. d := 0;
For i := 1 To length(S) Do
If (S[i] = 0) and (S[i] = 9) Then d := d + 1;
- C. d := 0;
For i := 1 To length(S) Do
If (S[i] >= '0') and (S[i] <= '9') Then d := d + 1;
- D. d := 0;
For i := 1 To length(S) Do
If (S[i] = '0') and (S[i] = '9') Then d := d + 1;

II. Phân tự luận

Phần cơ bản.

- 12.1.** Viết chương trình nhập vào một xâu bất kì và in ra màn hình xâu ngược lại.
- 12.2.** Viết chương trình nhập vào một xâu sau đó chuyển các kí tự thường của xâu đã nhập sang kí tự hoa.
- 12.3.** Viết chương trình nhập vào một xâu sau đó chuyển các kí tự hoa của xâu đã nhập sang kí tự thường.
- 12.4.** Một xâu gọi là Palindrom nếu nó trùng với xâu đảo của nó. Viết chương trình nhập vào một xâu bất kì và cho biết nó có phải là xâu Palindrom hay không?
- 12.5.** Viết chương trình nhập vào một xâu bất kì, in ra màn hình xâu sau khi đã xóa hết các kí tự trắng dư thừa. Các kí tự trắng dư thừa là các kí tự trắng đầu chuỗi, cuối chuỗi, hoặc hai kí tự trắng đi liền nhau ở giữa chuỗi.
- 12.6.** Viết chương trình nhập vào một xâu bất kì và cho biết xâu có bao nhiêu từ. (Mỗi từ là một dãy các kí tự khác kí tự trắng liên tiếp nhau).
- 12.7.** Viết chương trình nhập vào một xâu là họ và tên của một người nào đó, sau đó in ra màn hình tên của người đó.

Phần nâng cao

- 12.8.** Viết chương trình đổi số nguyên n thành số nhị phân.
- 12.9.** Viết chương trình nhập vào một xâu rồi hiển thị mỗi từ trên một hàng (theo thứ tự xuất hiện) đồng thời cho biết tổng số từ có trong xâu trên.
- 12.10.** Viết chương trình nhập vào hai xâu biểu diễn số nguyên. Thực hiện tính tổng của hai xâu đó như là việc tính tổng của hai số nguyên.
- 12.11.** Viết chương trình nhập vào từ bàn phím danh sách học sinh một lớp sau đó sắp xếp danh sách theo trật tự abc.
- 12.12.** Viết chương trình nhập vào hai từ rồi kiểm tra chúng có phải là anagram của nhau hay không? Hai từ là anagram của nhau nếu hoán vị của từ này sẽ ra từ kia. Ví dụ: dear và read là anagram của nhau.

Bài 13

KIỂU BẢN GHI (record)

A. PHƯƠNG PHÁP GIẢI TOÁN

1. Cách tạo kiểu bản ghi, khai báo biến kiểu bản ghi.

- Mỗi thông tin của đối tượng được gọi là một thuộc tính hay một trường. Mỗi đối tượng được mô tả bằng nhiều thông tin trên một hàng được gọi là một bản ghi.

- Để mô tả các đối tượng như vậy, ngôn ngữ lập trình cho phép ta xác định kiểu bản ghi. Mỗi đối tượng được mô tả bằng một bản ghi.

- Một ngôn ngữ lập trình luôn có một quy tắc để xác định: tên kiểu bản ghi, tên các trường, tên kiểu dữ liệu của mỗi trường, cách khai báo biến và cách tham chiếu đến từng trường.

- Khai báo kiểu bản ghi:

```
Type <tên_kiểu_bg> = Record  
    <Tên_trường_1>: <Kiểu_trường_1>;  
    .....  
    <Tên_trường_n>: <Kiểu_trường_n>;
```

End;

- Khai báo biến bản ghi: Var <tên_biến>: <tên_kiểu_bg>;

- Khai báo trực tiếp:

```
Var <tên_biến>: Record  
    <Tên_trường_1>: <Kiểu_trường_1>;  
    .....  
    <Tên_trường_n>: <Kiểu_trường_n>;
```

End;

2. Các thao tác trên biến kiểu bản ghi

+ Gán biến bản ghi cho biến bản ghi (cùng kiểu khai báo): Hai biến bản ghi có cùng kiểu dữ liệu có thể gán được cho nhau, khi đó giá trị dữ liệu trong các trường của biến thứ nhất lần lượt được gán vào cho các trường tương ứng của biến thứ hai.

+ Truy cập đến dữ liệu của các trường trong bản ghi:

Cú pháp 1: < Tên biến. tên trường >

- Nhập/xuất giá trị: Phải viết lệnh nhập/xuất lần lượt với từng trường. Write(Tên biến. tên trường), Read(Tên biến. tên trường).

Cú pháp 2: Sử dụng câu lệnh With <tên biến> do

Begin <Đọc/ghi tên trường> End;

Ví dụ: Viết chương trình nhập một danh sách nhân viên bao gồm các thông tin sau: Họ và tên, tuổi, địa chỉ, hệ số lương. Quá trình nhập sẽ kết thúc khi ta nhập họ và tên = ". Sắp xếp danh sách theo thứ tự tăng dần của hệ số lương và hiển thị danh sách sau khi sắp xếp ra màn hình. Hiển thị những người có hệ số lương >2.

Program Nhansu;

Type

Nhansu = Record

Hoten : String[30];

Tuoi : Byte;

Diachi : String[50];

HSL : Real;

End;

Var A : Array[1..50] Of Nhansu;

ht : String[30];

i : Byte;

Begin

n:=0;

Repeat

Write('Ho va ten: ');Readln(ht);

n := n+1;

IF ht <> " Then

With A[n] DO

Begin

Hoten := ht;

Write('Tuoi: ');Readln(tuoi);

Write('Dia chi :');Readln(diachi);

Write('He so luong: ');Readln(HSL);

End;

Until ht = ";

For i:=1 To n-1 Do

For j:=i+1 To n Do

If A[i].HSL > A[j].HSL Then

Begin

Tg := A[i];

A[i] := A[j];

A[j] := tg;

```

        End;
        Writeln('Danh sach sau khi sap xep');
        For i:= 1 To n Do
            With A[i] DO
                Begin
                    Writeln('Nguoi thu: ',i);
                    Writeln('Hò va ten: ',hoten);
                    Writeln('Tuoi: ',tuoi);
                    Writeln('Dia chi : ',diachi);
                    Writeln('He so luong: ',HSL);
                End;
            Writeln('Nhưng người có hệ số lương lớn hơn 2');
            For i:= 1 To n Do
                With A[i] Do
                    If HSL>2 Then
                        Begin
                            Writeln('Nguoi thu: ',i);
                            Writeln('Ho va ten: ',hoten);
                            Writeln('Tuoi: ',tuoi);
                            Writeln('Dia chi : ',diachi);
                            Writeln('He so luong: ',HSL);
                        End;
                End;
            Readln;
        End.

```

3. Kiểu tập hợp (Set type)

Khái niệm:

Dữ liệu kiểu tập hợp là một tập hợp của các dữ liệu cùng thuộc kiểu vô hướng hay miền con (là kiểu cơ sở), không được là kiểu số thực. Số phần tử tối đa của một tập hợp là 256.

Cách khai báo

- Khai báo gián tiếp:

```

Type Tên_kiểu = Set Of Kiểu_phần_tử;
Var biến : Tên_kiểu;

```

- Khai báo gián tiếp:

```

Var Biến: Set Of Kiểu_phần_tử;

```


Ví dụ:

```
Type chu_cai = Set Of 'a' .. 'z';  
Chu_so = Set Of 0..9;  
Var chu : chu_cai;  
So : chu_so;  
Mau : Set Of (xanh, do, tim, vang);
```

Chú ý:

- Các giá trị được đưa vào cần có số thứ tự trong khoảng từ 0 đến 255.
- Một dữ liệu kiểu tập hợp có dạng các phần tử nằm trong hai dấu ngoặc vuông []

Ví dụ: ['A','b','C'], [2,3,5,6]

- Vị trí các phần tử trong tập hợp không có ý nghĩa, tức [1,2]=[2,1]
- Tập hợp rỗng được kí hiệu là []
- Biến tập hợp cho phép có từ 0 đến 255 phần tử

Các phép toán trên tập hợp

- Phép gán: Có thực hiện các phép gán trên các biến kiểu tập hợp

Ví dụ: so:=[1,4,7,8]
 Chu:=['A','b','C']
 Rong:=[]

- Phép hợp: Được kí hiệu bằng dấu +. Hợp của hai tập là một tập có các phần tử thuộc hai tập.

Ví dụ:

```
A := [3..5];  
B := [4..9,15..20];  
C := A+B; (Tập C sẽ là [3..5,4..9,15..20])
```

- Phép giao: Được kí hiệu bằng dấu *. Giao của hai tập là một tập các phần tử thuộc A và thuộc B.

V dụ: C:= A*B; (C sẽ là [4,5])

- Phép hiệu: Được kí hiệu bằng dấu "- ". Hiệu của hai tập là một tập các phần tử thuộc tập thứ nhất nhưng không có trong tập thứ hai.

V dụ: C:= A-B; (C sẽ là [3])

- Phép thử "Thuộc về": Được kí hiệu bằng từ khoá IN. Là một phép thử để xem một biến hay một giá trị có thuộc một tập nào đó hay không, kết quả trả về là kiểu Boolean.

V dụ: If ch IN ['a'..'z', 'A'..'Z'] then writeln('ch là một kí tự');

- Các phép quan hệ:

+ Phép bằng : Cho giá trị True nếu 2 tập hợp bằng nhau.

- + Phép $\subsetneq$: Cho giá trị True nếu 2 tập hợp khác nhau.
- + Phép $\subseteq$: $A \subseteq B$ cho giá trị True nếu A là tập con của B.
- + Phép $\supseteq$: $A \supseteq B$ cho giá trị True nếu B là tập con của A.
- + Không có phép $>$, $<$ trong kiểu tập hợp. Nếu muốn so sánh lớn hơn hay nhỏ hơn ta có thể dùng thủ thuật sau:

If $(A \subseteq B)$ And $(A \subsetneq B)$ Then Writeln('A $\subset$ B');

Để thử xem hai tập hợp A và B có phần tử nào chung không ta dùng phép thử:

If $A * B = []$ Then Writeln('A và B không có phần tử nào chung');

Ví dụ: Viết chương trình lặp lại nhiều lần công việc sau; nhập một kí tự, nếu là chữ cái thì in ra thông báo 'La chu cai', nếu là chữ số thì in ra thông báo 'La chu so'. Nếu là kí tự khác thì kết thúc chương trình.

Uses CRT;

Var ch : Char;

Begin

Writeln('Nhap mot ky tu');

Repeat

Ch:=Readkey;

If ch IN ['a'..'z','A'..'Z'] Then Writeln('La mot ky tu')

Else

If ch IN ['0'..'9'] Then Writeln('La mot chu so')

Else Exit;

Until False;

Readln;

End.

B. BÀI TẬP ÁP DỤNG

I. Bài tập trắc nghiệm

Hãy chọn phương án trả lời đúng nhất.

Phần cơ bản

13.1. Để định nghĩa một kiểu bản ghi mới ta dùng cú pháp:

A. Type Tên kiểu = Record

Tên trường 1: Kiểu 1;

Tên trường 2: Kiểu 2;

....

Tên trường n: Kiểu n;

B. Var Tên kiểu = Record

Tên trường 1: Kiểu 1;
Tên trường 2: Kiểu 2;
....
Tên trường n: Kiểu n;

End;

C. var Tên kiểu = Record

Tên trường 1: Kiểu 1;
Tên trường 2: Kiểu 2;
....
Tên trường n: Kiểu n;

End.

D. A. Type Tên kiểu = Record

Tên trường 1: Kiểu 1;
Tên trường 2: Kiểu 2;
....
Tên trường n: Kiểu n;
End;

13.2. Để khai báo biến kiểu bản ghi trực tiếp ta sử dụng cú pháp:

A. Var Tên biến = Record

Tên trường 1: Kiểu 1;
Tên trường 2: Kiểu 2;
....
Tên trường n: Kiểu n;
End;

B. Var Tên biến : Record

Tên trường 1: Kiểu 1;
Tên trường 2: Kiểu 2;
....
Tên trường n: Kiểu n;
End;

C. Var <Tên biến> :< Tên kiểu> ;

D. Var Tên biến : Record

Tên trường 1: Kiểu 1;
Tên trường 2: Kiểu 2;
....
Tên trường n: Kiểu n;

13.3. Để khai báo biến kiểu bản ghi gián tiếp ta sử dụng cú pháp:

A. Type Tên kiểu = Record
 Tên trường 1: Kiểu 1;
 Tên trường 2: Kiểu 2;

 Tên trường n: Kiểu n;

Var Tên biến: Tên kiểu;

B. Var Tên kiểu = Record
 Tên trường 1: Kiểu 1;
 Tên trường 2: Kiểu 2;

 Tên trường n: Kiểu n;
End;

Var Tên biến: Tên kiểu.

C. Type Tên kiểu = Record
 Tên trường 1: Kiểu 1;
 Tên trường 2: Kiểu 2;

 Tên trường n: Kiểu n;
End;

Var Tên biến: Tên kiểu;

D. Const Tên kiểu = Record
 Tên trường 1: Kiểu 1;
 Tên trường 2: Kiểu 2;

 Tên trường n: Kiểu n;
End;

Var Tên biến: Tên kiểu;

13.4. Trong các khai báo sau khai báo nào đúng:

A. Type Hocsinh = Record
 Ten : string[30];
 SBD: Integer;
End;

Var hs: Hocsinh ;

B. Type Manghs : Record
 Ten : string[30];
 SBD: Integer;

```

        End;
    Vars: Mangs ;
C. Va M1, M2: Hocsinh;
D. Tye Manghs : Record
    Ten : string[30];
    SBD: Integer;
    End;

    Vars : Mang ;

```

13.5. Trong các khai báo sau khai báo nào sai?

- A. Typ Thisinh = Record
 Ten: string;
 Tuoi: Byte;
 End;
- B. Vars1: Record
 Ten: string;
 Tuoi: Byte;
 End;
- C. Vars1= Record
 Ten: string;
 Tuoi: Byte;
 End;
- D. Typ Nhanvien = Record
 Ten: String;
 Luong: Integer;
 End;
- Var M1: Nhanvien;

Phần ứng cao

13.6. Trong các phát biểu sau, phát biểu nào đúng?

- A Hai biến bản ghi cùng kiểu có thể gán được cho nhau.
 B Hai biến bản ghi có thể gán được cho nhau.
 C Biến bản ghi có thể nhận bất kì kiểu dữ liệu nào.
 D Các thao tác đọc, ghi biến bản ghi giống như các loại biến khác.

13.7. Để truy cập đến dữ liệu của một trường nào đó trên biến bản ghi ta sử dụng cú pháp:

- A Tên trường. Tên biến B. Tên biến. Tên trường
 C Tên trường. Tên biến D. Tên biến: Tên trường

13.8. Đề định nghĩa một kiểu bản ghi biến đổi ta dùng cú pháp nào?

A. Type Tênkiểu = Record

Trường 1: Kiểu 1;

Trường 2: Kiểu 2;

.....

Trường n: Kiểu n;

Case Trường thay đổi: Kiểu thay đổi

Giá trị 1: (Trường 11: Kiểu 11);

Giá trị 2: (Trường 22: Kiểu 22);

....

Giá trị m: (Trường mm: Kiểu mm);

End;

B. Type Tênkiểu = Record

Trường 1: Kiểu 1;

Trường 2: Kiểu 2;

.....

Trường n: Kiểu n;

Case Trường thay đổi: Kiểu thay đổi of

Giá trị 1: (Trường 11: Kiểu 11);

Giá trị 2: (Trường 22: Kiểu 22);

....

Giá trị m: (Trường mm: Kiểu mm);

End;

C. Type Tênkiểu = Record

Trường 1: Kiểu 1;

Trường 2: Kiểu 2;

.....

Trường n: Kiểu n;

Case Trường thay đổi: Kiểu thay đổi of

Giá trị 1: (Trường 11: Kiểu 11);

Giá trị 2: (Trường 22: Kiểu 22);

....

Giá trị m: (Trường mm: Kiểu mm);

D. Type Tênkiểu = Record

Trường 1: Kiểu 1;

Trường 2: Kiểu 2;

.....

Trường n: Kiểu n;

Trường thay đổi: Kiểu thay đổi

Giá trị 1: (Trường 11: Kiểu 11);

Giá trị 2: (Trường 22: Kiểu 22);

....

Giá trị m: (Trường mm: Kiểu mm);

13.9. Để mô tả một kiểu bản ghi Date gồm có ba trường: Ngày, tháng, năm ta sử dụng khai báo nào dưới đây:

A. Type Date: Record

Ngày: 1..31;

Tháng: 1..12;

Năm: Word;

End;

B. Var Date: Record

Ngày: 1..31;

Tháng: 1..12;

Năm: Word;

End;

C. Type Date = Record

Ngày: 1..31;

Tháng: 1..12;

Năm: Word;

End;

D. Type Date = Record

Ng: Ngay;

Th: Thang;

Nm: Nam;

End;

13.10. Để khai báo hai biến nguoi1, nguoi2 có kiểu bản ghi là Nhansu trong đó kiểu bản ghi nhansu bao gồm các trường ten, chucvu có kiểu string, trường luong có kiểu số thực. Khai báo nào dưới đây là đúng:

A. Type Nhansu = Record

Ten, chucvu: String;

Luong: Real;

End;

Var nguoi1, nguoi2: Nhansu;

B. Type Nhansu : Record

Ten, chucvu: String;

Luong: Real;

End;

Var nguoi1, nguoi2: Nhansu;

C. Var nguoi1, nguoi2: Record

Ten, chucvu: String;

Luong: Real;

End;

D. Cả hai đáp án A và C đúng.

13.11. Đề khai báo hai biến nguoi1, nguoi2 có kiểu bản ghi là Nhansu trong đó kiểu bản ghi nhansu bao gồm các trường ten, chucvu có kiểu string, trường luong có kiểu số thực, trường ngaysinh là một trường có kiểu bản ghi Date gồm các trường sau: Ngay có giá trị từ 1..31, trường thang có giá trị từ 1..12, trường nam có giá trị kiểu Word. Khai báo nào dưới đây là đúng:

A. Type Nhansu = Record

Ten, chucvu: String;

Luong: Real;

Ngaysinh: Date;

End;

Var nguoi1, nguoi2: Nhansu;

B. Type Date = Record

Ngay: 1..31;

Thang: 1..12;

Nam: Word;

End;

Type Nhansu : Record

Ten, chucvu: String;

Luong: Real;

Ngaysinh: Date;

End;

Var nguoi1, nguoi2: Nhansu;

C. Var nguoi1, nguoi2: Record

Ten, chucvu: String;

Luong: Real;

End;

D. Cả hai đáp án A và B đúng.

13.12. Giả sử biến Ts có kiểu Thisinh bao gồm các trường ten, sobaodanh, diem đã được định nghĩa và khai báo. Cách viết nào dưới đây là đúng:

A. Ts.ten := "nguyen van an";

 Readln(ts.sobaodanh);

 Readln(diem);

B. Readln(ten);

 Readln(sobaodanh);

 Readln(diem);

C. With Ts do

 Begin

 Readln(ten);

 Readln(sobaodanh);

 Readln(diem);

 End;

D. Cả hai đáp án A và C đúng.

13.13. Trong ngôn ngữ lập trình Pascal, để khai báo kiểu bản ghi để xử lý danh sách cán bộ của một cơ quan, khai báo kiểu bản ghi nào trong các khai báo sau là đúng? (Bản ghi có cấu trúc gồm 5 trường: Họ tên, năm sinh, địa chỉ, mức lương, phụ cấp)

A. Type CanBo = Record

 HoTen : String[30];

 NamSinh : Integer;

 DiaChi : String[100];

 MucLuong : Real;

 PhuCap : Real;

End;

B. Type CanBo : Record

 HoTen : String[30];

 NamSinh : Integer;

 DiaChi : String[100];

 MucLuong : Real;

 PhuCap : Real;

End;

C. Type CanBo = Record

 HoTen : String[30];

 NamSinh : Integer;

 DiaChi : String[100];

```

MucLuong : Real;
PhuCap : Real;
D. Type CanBo := Record
    HoTen : String[30];
    NamSinh : Integer;
    DiaChi : String[100];
    MucLuong : Real;
    PhuCap : Real;
End.

```

13.14. Câu lệnh nào trong các câu lệnh sau không dùng để gán giá trị cho trường của bản ghi A? (Với bản ghi A có 3 trường là Hoten, Lop, Diem)

- A. A.Ten := 'Nguyen Van A';
- B. A.Lop := '10A7';
- C. Readln(A.Diem);
- D. S := A.Diem;

13.15. Trong kiểu dữ liệu bản ghi, mỗi bản ghi thường được dùng để?

- A. Mô tả hay lưu trữ thông tin về nhiều đối tượng cần quản lí
- B. Mô tả hay lưu trữ thông tin về một thuộc tính cần quản lí
- C. Mô tả hay lưu trữ thông tin về một đối tượng cần quản lí
- D. Mô tả hay lưu trữ thông tin về nhiều thuộc tính cần quản lí

13.16. Một kiểu bản ghi được khai báo như sau:

```

Type Sach = Record
    Ten_sach : String[30];
    Tac_gia : Byte;
    Gia_tien : Real;
End;

```

Var X, Y, Z : Sach;

Câu lệnh nào trong các câu lệnh sau là không phù hợp?

- A. Write('X=', X, 'Y=', Y, 'Z=', Z);
- B. Writeln(X.Ten_sach, Y.Tac_gia, Z.Gia_tien);
- C. X := Y;
- D. Y := Z;

II. Bài tập tự luận

Phần cơ bản

13.1. Khai báo mô tả Record Diachi có ba trường số nhà, tên đường, thành phố có kiểu dữ liệu phù hợp.

13.2. Mỗi phân số được lưu trong một bản ghi gồm có hai trường đó là tử số và mẫu số. Hãy định nghĩa kiểu phân số sau đó viết chương trình thực hiện các công việc cộng, trừ, nhân, chia hai phân số.

13.3. Để quản lý việc tiêu thụ điện của các hộ gia đình, mỗi khách hàng được sở điện lực lưu trong một bản ghi gồm có các trường như sau: Họ tên, địa chỉ, chỉ số điện cũ, chỉ số điện mới. Viết chương trình tính tiền tiêu thụ điện hàng tháng của khách hàng biết rằng số điện tiêu thụ bằng chỉ số điện mới trừ chỉ số điện cũ, 20 chỉ số đầu tiên có giá là 500 đồng, 30 chỉ số tiếp theo có giá là 700 đồng và số còn lại có giá 1000 đồng.

13.4. Để quản lý điểm học tập của học sinh phổ thông, mỗi môn học người ta quản lý các thông tin sau:

- Điểm kiểm tra miệng có hệ số 1, gồm 1 cột điểm.
- Điểm kiểm tra 15 phút có hệ số 1, gồm 2 cột điểm.
- Điểm kiểm tra 1 tiết có hệ số 2, gồm 2 cột điểm.
- Điểm kiểm tra học kì gồm 1 cột điểm.
- Điểm trung bình kiểm tra (TBKT) là điểm trung bình của các điểm kiểm tra.

- Điểm trung bình môn được tính như sau: $TBM = (TBKT * 2 + \text{điểm học kì}) / 2$.

Viết chương trình nhập danh sách học sinh và các cột điểm tương ứng. Thông báo lên màn hình danh sách học sinh phải thi lại môn học này (học sinh thi lại là học sinh có điểm trung bình dưới 5).

13.5. Viết chương trình nhập họ tên và điểm trung bình của một lớp. In danh sách theo điểm trung bình giảm dần và có xếp hạng.

13.6. Để quản lý kết quả thi tuyển vào một trường Đại học mỗi thí sinh cần lưu giữ các thông tin sau: Số báo danh, họ và tên, điểm thi các môn toán, lí, hoá. Viết chương trình thực hiện công việc sau:

- a. Nhập vào danh sách thí sinh tham gia dự thi và kết quả.
- b. Sắp xếp theo thứ tự alphabet của danh sách thí sinh đã nhập vào.
- c. Hiện thị danh sách thí sinh lên màn hình kèm theo điểm các môn và tổng điểm.
- d. Hiện thị danh sách thí sinh trúng tuyển kèm theo điểm các môn và tổng điểm. Thí sinh trúng tuyển là thí sinh có tổng điểm từ 18 điểm trở lên và không có môn nào có điểm dưới 3.

13.7. Để quản lý giáo viên của một trường học người ta quản lý giáo viên với các thông tin sau: Họ tên, năm công tác, lương, giới tính. Viết chương trình thực hiện các công việc sau:

- a. Nhập vào một danh sách giáo viên, yêu cầu việc nhập được tiến hành khi kiểm tra họ tên khác rỗng mới cho nhập các thông tin khác.
 - b. Sắp xếp danh sách theo trật tự giảm dần của lương.
 - c. Thông báo lên màn hình danh sách các giáo viên đến tuổi nghỉ hưu, cho biết giáo viên nam có số năm công tác lớn hơn 35 năm và giáo viên nữ có số năm công tác lớn hơn 32 năm thì được nghỉ hưu.
- 13.8.** Để quản lí xe máy trong một cửa hàng người ta quản lí các thông tin sau: Kí hiệu xe, tên xe, nguồn gốc, giá tiền, thuế. Viết chương trình thực hiện các công việc sau:
- a. Nhập vào danh sách các xe máy có trong cửa hàng, trong đó thuế không nhập vào mà được tính theo công thức thuế bằng $10\% \cdot \text{giá tiền}$.
 - b. Tìm và hiển thị các xe máy có nguồn gốc từ Nhật Bản.
 - c. Cho biết tên xe, kí hiệu xe và màu của xe máy có giá cao nhất.
- 13.9.** Đường thẳng đi qua điểm (x_1, y_1) với hệ số góc là m có thể biểu diễn bằng phương trình $y - y_1 = m(x - x_1)$.
- a. Dùng bản ghi để mô tả đường thẳng bằng toạ độ của một điểm và hệ số góc.
 - b. Viết chương trình đọc toạ độ và hệ số góc của hai đường thẳng rồi xác định chúng có cắt nhau hay song song với nhau. Nếu chúng cắt nhau, tìm giao điểm và xác định chúng có vuông góc với nhau hay không.
- 13.10.** Một phân số có thể biểu diễn bằng bản ghi gồm tử số và mẫu số. Viết chương trình thực hiện các công việc sau:
- Nếu nhập vào một phân số thì chương trình hiển thị phân số đảo ngược, phân số rút gọn và dạng hợp số nếu tử số lớn hơn mẫu số.
 - Nếu nhập vào hai phân số thì hiển thị bội chung nhỏ nhất của hai mẫu số, tổng của chúng dưới dạng rút gọn và quan hệ giữa hai phân số đó như là bằng nhau, lớn hơn hay bé hơn.

Chương 5.

TỆP VÀ THAO TÁC VỚI TỆP

Nội dung chủ yếu của chương là:

- *Phân loại tệp.*
- *Khai báo biến tệp, thao tác với tệp: Gán tên tệp, mở/đóng tệp, đọc/ghi tệp.*

Bài 14

Kiểu dữ liệu tệp

A. PHƯƠNG PHÁP GIẢI TOÁN

1. *Khái niệm về tệp (file)*

Khi thực hiện chương trình, các kiểu dữ liệu như mảng, xâu, bản ghi,... là các cấu trúc dữ liệu được tổ chức lưu trữ trên bộ nhớ RAM của máy tính nên khi kết thúc chạy chương trình hoặc mất điện thì các dữ liệu này cũng mất luôn. Đối với một số bài toán ta muốn dùng lại dữ liệu nhiều lần thì buộc phải tổ chức lưu trữ trên đĩa bằng kiểu lưu trữ tệp. Với cách lưu trữ này, khi kết thúc chương trình thì dữ liệu vẫn tồn tại trên đĩa và các dữ liệu đó vẫn có thể sử dụng lần sau.

Tệp hay tệp dữ liệu là một tập hợp các dữ liệu có liên quan với nhau có cùng kiểu được nhóm lại với nhau và được lưu trữ ở bộ nhớ ngoài như đĩa cứng, đĩa mềm với một tên nào đó do ta đặt.

2. *Đặc điểm của kiểu tệp*

- + Được lưu trữ lâu dài ở bộ nhớ ngoài, không bị mất khi mất điện.
- + Lượng thông tin lưu trữ trên tệp có thể rất lớn.
- Có hai loại tệp:
 - + Tệp có cấu trúc là loại tệp mà các thành phần của nó được tổ chức theo một cấu trúc nhất định.
 - + Tệp văn bản: là tệp mà dữ liệu được ghi dưới dạng các kí tự theo mã ASCII. Trong tệp văn bản, dãy kí tự kết thúc bởi kí tự xuống dòng hay kí tự kết thúc tệp tạo thành một dòng.

3. *Cấu trúc và phân loại tệp*

Các phần tử của một Record hay một Array có thể truy nhập được tùy ý (Random access) thông qua tên biến, chỉ dẫn hoặc tên trường. Các phần tử của tệp không có tên và việc truy nhập không thể tùy tiện được. Các phần tử

của tệp được sắp xếp thành một dãy và ở mỗi thời điểm chương trình chỉ có thể truy nhập vào một phần tử của tệp thông qua giá trị của một biến đếm.

Ta có thể hình dung ra một tệp như là một cuộn phim chụp ảnh. Mỗi một ảnh là một phần tử và ống kính là cửa sổ để nhìn vào nên tại mỗi thời điểm chỉ có thể nhìn thấy 1 ảnh. Sau mỗi lần chụp, cửa sổ sẽ nhìn vào ảnh ở vị trí tiếp theo.

Mỗi tệp đều được kết thúc bằng một dấu hiệu đặc biệt để báo hiệu hết tệp (End Of File).

Việc phân loại tệp dựa trên việc bố trí các phần tử của tệp trong bộ nhớ ngoài và cách truy nhập vào tệp:

- + Tệp truy nhập tuần tự (Sequential access): Việc đọc một phần tử bất kì của tệp phải đi qua các phần tử trước đó. Muốn ghi thêm một phần tử mới vào tệp phải đặt cửa sổ tệp về cuối tệp.

- + Tệp truy nhập trực tiếp (Direct access): Có thể truy nhập tới bất kì phần tử nào trong tệp thông qua chỉ số thứ tự của phần tử trong tệp.

B. BÀI TẬP ÁP DỤNG

I. Bài tập trắc nghiệm

Hãy chọn phương án trả lời đúng nhất.

Phần cơ bản

14.1. Xét theo cách tổ chức dữ liệu kiểu dữ liệu tệp (File) có 2 loại đó là:

- A. Tệp có cấu trúc và tệp văn bản.
- B. Tệp có cấu trúc và tệp định kiểu.
- C. Tệp định kiểu và tệp số.
- D. Tệp văn bản và tệp số.

14.2. Sự khác nhau cơ bản về mặt lưu trữ trong bộ nhớ giữa kiểu dữ liệu tệp và các kiểu dữ liệu khác là:

- A. Kiểu tệp lưu trữ trong bộ nhớ RAM, các kiểu khác lưu trong bộ nhớ ngoài.
- B. Kiểu tệp lưu trữ trong bộ nhớ ngoài (trên đĩa) còn các kiểu khác lưu ở bộ nhớ RAM.
- C. Kiểu tệp lưu trữ trong bộ nhớ ROM còn các kiểu khác lưu ở bộ nhớ RAM.
- D. Không khác nhau.

14.3. Xét theo cách thức truy cập dữ liệu kiểu tệp có hai loại đó là:

- A. Tệp truy cập tuần tự và tệp truy cập trực tiếp.
- B. Tệp truy cập tuần tự và tệp truy cập ngẫu nhiên.

C. Tập truy cập trực tiếp và tập định kiểu.

D. Tập truy cập tuần tự và tập định kiểu.

14.4. Trong kiểu dữ liệu file để định nghĩa một kiểu tệp mới ta sử dụng từ khoá nào dưới đây?

A. Uses.

C. Type.

B. Const.

D. Var.

14.5. Trong kiểu dữ liệu file để khai báo một biến mới ta sử dụng từ khoá nào dưới đây?

A. Type.

C. Program.

B. Begin.

D. Var.

II. Bài tập tự luận

14.1. Trình bày vai trò của dữ liệu kiểu tệp.

14.2. Có mấy tiêu chí để phân loại dữ liệu kiểu tệp, hãy trình bày các đặc điểm cơ bản của các tiêu chí đó.

14.3. Trình bày các thao tác cơ bản đối với một kiểu dữ liệu.

14.4. Kiểu dữ liệu tệp có những đặc điểm nào?

Bài 15

THAO TÁC VỚI TỆP

A. PHƯƠNG PHÁP GIẢI TOÁN

1. Cách khai báo

Tệp là một kiểu dữ liệu có cấu trúc. Định nghĩa của tệp có phần nào giống mảng ở chỗ chúng đều là tập hợp của các phần tử dữ liệu có cùng kiểu. Song mảng được định nghĩa và khai báo trong chương trình với số phần tử đã xác định còn số phần tử của tệp không được xác định khi định nghĩa.

- Cách khai báo gián tiếp

Type Kiểu_tệp = FILE OF Kiểu_Ptử;

Var Biến_tệp : Kiểu_tệp;

- Cách khai báo trực tiếp

Var Biến_tệp : FILE OF Kiểu_Ptử;

Ví dụ:

Type

FileInteger = FILE OF Integer;


```

FileReal = FILE OF Real;
Nhansu = Record
.   Hoten : String[30];
    Tuoi : Byte;
    Diachi : String[40];
    Luong : Real;
End;
FileNhansu = FILE OF Nhansu;
Var
  F1 : FileInteger;
  F2 : FileReal;
  F3 : FileNhansu;
  F4 : FILE OF Char;
  F5 : FILE OF Array[1..50] OF Nhansu;

```

2. Các thủ tục và hàm thao tác trên tệp

- *Mở tệp mới để ghi:*

```

Assign(Biến_tệp,Tên_tệp);
Rewrite(Biến_tệp);

```

Trong đó: Thủ tục Assign: Gán tên_tệp cho một biến_tệp. Với tên_tệp là một xâu kí tự chỉ đường dẫn đến tệp trên đĩa.

Ví dụ: Assign(F1,"C:\songuyen.Dat");
 Rewrite(F1);

Chú ý: Sau khi mở xong tệp sẽ rỗng và khi sử dụng thủ tục Rewrite nếu trên bộ nhớ ngoài đã có sẵn tệp có trùng tên với tệp được mở thì tệp cũ sẽ bị xoá đi.

- *Mở tệp đã tồn tại để đọc:*

```

Assign(Biến_tệp,Tên_tệp);
Reset(Biến_tệp);

```

Ví dụ: Assign(F1,"C:\songuyen.Dat");
 Reset(F1);

Chú ý: Sau khi sử dụng thủ tục Reset, nếu tệp không rỗng thì cửa sổ tệp trở vào phần từ đầu tiên của tệp.

- *Đóng tệp đã mở: Close(Biến_tệp);*

Ví dụ: Close(F1);

- *Ghi dữ liệu vào tệp:*

```

Write(Biến_tệp,Item_1,Item_2,...,Item_n);

```

Trong đó: Item_i có thể là các hằng, các biến, các biểu thức và tất nhiên các Item_i phải có giá trị cùng với kiểu phần tử của tệp.

Sau khi ghi dữ liệu xong của sổ tệp sẽ tự động di chuyển sang vị trí mới.

Ví dụ: Write(F1,4,i*4,i);

- *Đọc dữ liệu từ tệp:*

Read(Biến_tệp,var_1,var_2,...,var_n);

Thủ tục sẽ đọc các giá trị tại vị trí của sổ đang trỏ và gán cho biến var_i tương ứng cùng kiểu. Sau đó dịch chuyển của sổ sang vị trí tiếp theo và đọc giá trị cho biến khác (nếu có).

Ví dụ: Read(F1,x,y,z);

Chú ý: Trước khi đọc một phần tử của tệp thì cần phải kiểm tra xem tệp có còn phần tử nào hay không tức là của sổ tệp đã trở đến cuối tệp chưa (End Of File). Để kiểm tra xem của sổ tệp đã ở cuối tệp hay chưa, ta sử dụng hàm EOF kiểu Boolean: EOF(biến_tệp)=True nếu của sổ tệp trở vào cuối tệp, còn nếu của sổ tệp chưa trở vào phần tử cuối cùng của tệp thì EOF(biến_tệp)=False

Khi đọc tệp ta cần kiểm tra xem đã kết thúc tệp hay chưa bằng câu lệnh:

If NOT EOF(Biến_tệp) Then Read(biến_tệp,x);

Hoặc

While NOT EOF(biến_tệp) Do

Begin

Read(biến_tệp,x);

...

End;

Ví dụ: Đọc dữ liệu từ tệp đã tạo ở trên và hiển thị ra màn hình

Program Doc_tep;

Type FI = File Of Integer;

Var f1 : FI;

N : Integer;

St : String;

Begin

Write('ten tep: '); Readln(st);

Assign(f,st);

Reset(f);

While NOT EOF(f) Do

Begin

```

        Read(f,x);
        Write(x);
    End;
    Close(f);
    Readln;
End.

```

3. Một số thủ tục và hàm thường dùng

- Thủ tục Seek(biến_tệp, n): Chuyển cửa sổ tệp vào phần tử thứ n (phần tử đầu tiên của tệp được đánh số là 0).

- Hàm FileSize(biến_tệp): Cho số phần tử của tệp. Filesize cho giá trị 0 khi tệp rỗng.

Để đưa con trỏ tệp tới phần tử cuối cùng của tệp ta dùng lệnh:

```
Seek(biến_tệp, FileSize(biến_tệp));
```

- Hàm FilePos(biến_tệp): Cho vị trí hiện thời của cửa sổ tệp.

- Thủ tục Erase(Biến_tệp): Xoá file

- Thủ tục Rename(Biến_tệp, tên_tệp_mới): Đổi tên tệp

4. Tệp văn bản (Text Files)

4.1. Khái niệm

Trong Pascal có một kiểu tệp đã được định nghĩa trước, đó là tệp văn bản được định nghĩa với từ chuẩn Text.

Các phần tử của tệp kiểu Text là các kí tự. File Text được tổ chức thành từng dòng với độ dài mỗi dòng khác nhau nhờ có thêm dấu hiệu hết dòng (End of line).

Ví dụ: Khai báo các biến tệp F1, F2 có kiểu Text:

```
Var F1, F2 : Text;
```

4.2. Các thao tác trên tệp văn bản

Các thủ tục và hàm đã trình bày đối với tệp có cấu trúc cũng sử dụng được đối với tệp văn bản, ngoài ra tệp văn bản còn có thêm một số thủ tục và hàm sau:

- Ghi vào tệp:

```
Write(biến_tệp, Item1, Item2, ..., Itemn); (1)
```

```
Writeln(biến_tệp, Item1, Item2, ..., Itemn); (2)
```

```
Writeln(biến_tệp); (3)
```

Trong đó:

(1): Ghi các giá trị của các Itemi vào tệp.

(2): Thêm dấu hết dòng vào tệp sau khi đã viết hết các giá trị của

các Item i.

(3): Thêm dấu hết dòng vào tệp.

- Đọc dữ liệu từ tệp văn bản: Chúng ta có thể đọc không những các kí tự từ tệp văn bản mà còn có thể đọc lại các số nguyên, số thực, logic,... từ tệp văn bản thông qua các thủ tục sau:

Read(Biến_tệp, var1, var2, ..., varn): (1)

Readln(Biến_tệp, var1, var2, ..., varn); (2)

Readln(biến_tệp); (3)

Tong đó: Vari là các biến thuộc kiểu kí tự, nguyên, thực, logic, chuỗi.

(1): Đọc giá trị của các phần tử và gán cho các biến vari tương ứng mà không chuyển cửa sổ tệp xuống dòng.

(2): Sẽ đưa cửa sổ tệp sang đầu dòng tiếp theo sau khi đã lần lượt đọc các phần tử cho các biến tương ứng.

(3): Sẽ đưa cửa sổ tệp sang đầu dòng tiếp theo mà không làm gì cả.

- Thủ tục thêm dòng:

Append(biến_tệp); Mở tệp văn bản để ghi bổ sung thêm một dòng.

Chú ý:

+ Hàm EOF(Biến_tệp): Tương tự tệp có kiểu.

+ Hàm EOLN(biến_tệp) = True nếu cửa sổ tệp ở cuối dòng, và cho giá trị bằng FALSE trong trường hợp ngược lại.

+ Hàm SEEKEOLN(Biến_tệp): Có kiểu Boolean, tương tự EOLN song trước khi thử EOLN nó nhảy qua các dấu cách và Tab.

+ Hàm SEEKEOF(Biến_tệp): Có kiểu Boolean, tương tự EOF song trước khi thử EOF nó nhảy qua các dấu cách, Tab và dấu kết thúc dòng.

+ Không sử dụng SEEK, FILESIZE, FILEPOSE cho tệp văn bản.

B. BÀI TẬP ÁP DỤNG

I. Bài tập trắc nghiệm

Hãy chọn phương án trả lời đúng nhất.

Phần cơ bản

15.1. Cơ pháp nào sau đây để định nghĩa một kiểu tệp mới?

- A. Type Tên kiểu = File Of Kiểu phần tử;
- B. Var Tên kiểu = File Of Kiểu phần tử;
- C. Type Danh sách các biến: Tên kiểu phần tử;
- D. Tên kiểu = File Of Kiểu phần tử;

- 15.2.** Trong Kiểu dữ liệu tệp cú pháp nào dưới đây dùng để khai báo biến?
- A. Var Danh sách các biến: File of Kiểu phần tử;
 - B. Var Danh sách các biến :Tên kiểu; Trong đó tên kiểu đã được định nghĩa bằng từ khoá Type trước.
 - C. Var Danh sách các biến : Type Danh sách kiểu;
 - D. Cả a và b đều đúng.
- 15.3.** Để khai báo biến tệp dạng văn bản ta sử dụng cú pháp nào?
- A. Var Danh sách biến: File of;
 - B. Var Danh sách các biến : Text;
 - C. Var Danh sách biến: File;
 - D. Type Danh sách biến : Text;
- 15.4.** Trong các khai báo sau, khai báo nào đúng?
- A. Var file1, file2: Text;
 - B. Type Songuyen = File of Text;
 - C. Var file1, file2: File of Text;
 - D. Var f1, f2: File of Banghi;
- 15.5.** Trong các khai báo sau, khai báo nào không đúng?
- A. Var f: Text;
 - B. Type FSonguyen = File of Integer;
 - C. Type FSonguyen = File of Integer;
Var h: Fsonguyen;
 - D. Var f1, f2: File of Banghi;
- 15.6.** Trong tập tin dạng văn bản thủ tục nào dưới đây được sử dụng để gán tên file cho biến?
- A. Write(Biến file).
 - B. Rewrite(Biến file).
 - C. Assign(Biến file, tên file).
 - D. Assign(Biến file).
- 15.7.** Trong tập tin dạng văn bản thủ tục nào dưới đây được sử dụng để tạo mới một tệp trên đĩa?
- A. Assign(Biến file).
 - B. Erase(Biến file).
 - C. Rewrite(Biến file).
 - D. Rewrite(Biến file, tên file).
- 15.8.** Trong tập tin dạng văn bản thủ tục nào dưới đây được sử dụng để mở một file đã có trên đĩa?
- A. Close(Biến file).
 - B. Assign(Biến file).
 - C. Open(Biến file).
 - D. Reset(Biến file).
- 15.9.** Trong tập tin dạng văn bản thủ tục nào dưới đây được sử dụng để đóng một file đang mở?
- A. Erase(Biến file).
 - C. Close(Biến file).

B. Rename(Biến file).

D. EOF(Biến file).

15.10. Trong tập tin dạng văn bản thủ tục nào dưới đây được sử dụng để đổi tên một file?

A. Ren(Biến file, k).

C. Rename(Tên cũ, Tên mới).

B. Rename(Biến file, tên file mới).

D. Rename(Tên mới).

Phần nâng cao

15.11. Trong tập tin dạng văn bản thủ tục nào dưới đây được sử dụng để xóa một file đã có?

A. Delete(Biến file).

C. Erase(Biến file).

B. Erasing(Biến file).

D. Rewrite(Biến file).

15.12. Trong tập tin định kiểu hàm Filesize(Biến file) được sử dụng để:

A. Lấy giá trị số phần tử trong file có tên đã gán cho biến file.

B. Lấy số hiệu số phần tử của file có tên đã gán cho biến file.

C. Trả về kích thước của file có tên đã gán cho biến file.

D. Trả về kích thước của một biến trong file.

15.13. Trong tập tin định kiểu hàm Filepos(Biến file) được sử dụng để:

A. Trả về giá trị True nếu file tồn tại, giá trị False nếu file không tồn tại.

B. Trả về vị trí của file trong bộ nhớ.

C. Trả về vị trí của File mà cửa sổ file đang định vị với tên file đã gán cho biến file đang mở.

D. Trả về vị trí của phần tử mà cửa sổ file đang định vị của File có tên đã gán cho biến file đang mở.

15.14. Phát biểu nào dưới đây là sai?

A. Thủ tục Seek(Biến file, k) dùng để đưa cửa sổ file đến định vị vào phần tử có số hiệu k của tập tin có tên đã gán cho Biến file đang được mở.

B. Thủ tục Read(Biến file, biến) dùng để đọc dữ liệu tại phần tử mà cửa sổ file đang định vị của tập tin đang có tên đã gán cho biến file.

C. Thủ tục Write(Biến file, biểu thức) dùng để ghi giá trị từ biểu thức vào biến file.

D. Thủ tục EOF(Biến file) trả về giá trị True nếu cửa sổ file đang ở vị trí kết thúc file và trả về giá trị False nếu cửa sổ file chưa ở vị trí kết thúc.

15.15. Trong tập tin định kiểu, phát biểu nào dưới đây là đúng?

A. Muốn xóa một file bằng thủ tục Erase(biến file) thì file cần xóa phải ở trạng thái đóng.

- B. Muốn đổi tên một file bằng thủ tục Rename(biến file, tên mới) thì file đó phải ở trạng thái mở.
- C. Thủ tục Writeln(Biến file, biểu thức) dùng để ghi giá trị từ biểu thức vào biến file.
- D. Thủ tục Readln(Biến file, biến) dùng để đọc dữ liệu tại phần tử mà cửa sổ file đang định vị của tập tin đang có tên đã gán cho biến file, dữ liệu đọc ra và lưu vào biến file.
- 15.16.** Câu lệnh nào dưới đây dùng để gán tên file “solieu.dat” cho biến file f1 ?

A. F1:=Assign('solieu.dat'); C. Assign(F1, 'solieu.dat');
B. Assign(f1,solieu.dat); D. Assign(F1, “solieu.dat”);

- 15.17.** Giả sử tại thư mục hiện thời có tập tin DULIEU.DAT. Đoạn chương trình sau đây cho kết quả là gì ?

Var f: Text;

.....

Assign(f, 'DULIEU.DAT');
Rename(f, 'LUUTRU.DAT')

.....

- A. Mở một file mới có tên là DULIEU.DAT.
B. Đóng file có tên là DULIEU.DAT.
C. Đổi tên file DULIEU.DAT thành file có tên LUUTRU.DAT.
D. Đọc dữ liệu từ file DULIEU.DAT.

- 15.18.** Giả sử tại thư mục hiện thời có tập tin Baitap.inp. Đoạn chương trình sau đây làm gì ?

Var f: Text;

.....

Assign(f, 'Baitap.inp');
Erase(f);

.....

- A. Mở file Baitap.inp để đọc. C. Đóng file Baitap.inp.
B. Mở file Baitap.inp để ghi. D. X Xoá file Baitap.inp.

- 15.19.** Khi thực hiện đoạn chương trình sau:

Var f: Text;

Begin

Rewrite(f);

Close();

End.

thì máy sẽ báo lỗi. Nguyên nhân gây ra lỗi là do?

- A. Biến f chưa được khai báo.
- B. Chưa khai báo thư viện CRT.
- C. Chưa sử dụng thủ tục Assign để gán tên cho biến file.
- D. Chưa định nghĩa kiểu dữ liệu Text.

15.20. Các phần tử của tập tin lần lượt được gán số hiệu theo quy định là:

- A. 1, n-1, n-2, ..., 2, 1.
- B. , 2, 3, ..., n-2, n-1, n.
- C. 0, 1, 2, ..., n-1.
- D. n-1, n-2, ..., 2, 1, 0.

II. Bài tập tự luận

Phần cơ bản.

15.1. Viết cú pháp để khai báo một kiểu file có kiểu phần tử là các số nguyên.

15.2. Viết cú pháp để khai báo biến f1, f2 có kiểu dữ liệu file là file dạng văn bản.

15.3. Viết cú pháp để khai báo biến hs1, hs2 có kiểu dữ liệu file là file có kiểu phần tử là các bản ghi học sinh, trong đó mỗi học sinh bao gồm các thông tin sau: Số báo danh, họ tên và ngày sinh.

15.4. Viết chương trình tạo một tệp chứa các số nguyên từ 1 đến 100 với tên tệp là songuyen.dat.

15.5. Viết chương trình tạo một tập tin mới chứa các số nguyên có tên là songuyen.dat lưu tại thư mục hiện hành, sau đó thực hiện sắp xếp các số nguyên trong file theo trật tự giảm dần.

15.6. Viết chương trình để sao chép một tập tin dạng văn bản đã có trên đĩa thành một tập tin mới. Tên tập tin cũ và mới được nhập từ bàn phím.

15.7. Giả sử có một tệp chứa 100 số nguyên từ 1 đến 100. Viết chương trình kiểm tra xem phần tử thứ 2 (đếm từ 0) của tệp có giá trị bằng 3 hay không, nếu không bằng 3 thì phải sửa lại.

Bài 16

VÍ DỤ LÀM VIỆC VỚI TỆP

I. PHƯƠNG PHÁP GIẢI TOÁN

Xem lại các nội dung:

+ Tệp có cấu trúc:

- Gán tên tệp, mở tệp, tạo tệp mới, đóng tệp.

- Đọc/ghi tệp văn bản.

- Các hàm và thủ tục liên quan.

+ Tệp văn bản

- Gán tên tệp.

- Mở tệp.

- Tạo tệp mới.

- Đọc/ghi thông tin của tệp.

- Đóng tệp.

- Các hàm và thủ tục liên quan.

TÓM TẮT

- Khai báo tệp định kiểu :

VAR <Tên biến tệp> : **FILE OF** <Kiểu phần tử tệp>;

- Khai báo tệp văn bản :

VAR <Tên biến tệp>: **TEXT**;

- Gán tên tệp :

ASSIGN(< Tên biến tệp>, <Tên tệp>);

- Mở tệp :

○ Để đọc : **RESET**(<Tên biến tệp>);

○ Để ghi : **REWRITE**(<Tên biến tệp>);

- Đọc/ghi tệp :

○ Đọc : **READ**(<Tên biến tệp>, <Biến nhận>);

○ Ghi : **WRITE**(<Tên biến tệp>, <Biến đưa ra>);

- Sau khi làm việc xong phải đóng tệp bằng câu lệnh :

CLOSE(<Tên biến tệp>);

- Tệp văn bản :

Gán tên tệp, mở tệp và đóng tệp : như tệp định kiểu.

Truy nhập : như cách làm việc với bàn phím và màn hình.

Ví dụ 1

Lập trình đọc từ bàn phím số học sinh của lớp và kết quả kiểm tra môn Tin học của các học sinh trong lớp, ghi kết quả ra tệp KQ.DAT, mỗi phần tử của tệp là một bản ghi bao gồm Họ và Tên (xâu không quá 28 kí tự) và điểm Tin (kiểu BYTE).

```
PROGRAM KQ_TIN;
USES CRT;
TYPE BANGHI = RECORD
    HT : STRING[28];
    DIEM : BYTE
END;
VAR  N : INTEGER;
    BG : BANGHI;
    F : FILE OF BANGHI;
    TENTEP : STRING[80];
    I : INTEGER;
BEGIN
    CLRSCR;
    WRITE(' So hoc sinh : ');
    READLN(N);
    TENTEP := 'KQ.DAT';
    ASSIGN(F,TENTEP);
    REWRITE(F);
    FOR I := 1 TO N DO
        BEGIN
            WITH BG DO
                BEGIN
                    WRITE(' Ho Va Ten : ');
                    READLN(HT);
                    WRITE('DIEM : ');
                    READLN(DIEM)
                END;
            WRITE(F,BG)
        END;
    CLOSE(F)
END.
```

Ví dụ 2

Lập trình đọc tệp KQ.DAT ở ví dụ 1, tính và đưa ra màn hình điểm trung bình môn Tin học của lớp (làm tròn đến chữ số thập phân thứ nhất).


```

PROGRAM VIDU_2;
USES CRT;
TYPE BANGHI = RECORD
    HT : STRING[28];
    DIEM : BYTE
END;
VAR F : FILE OF BANGHI;
    N,TG : INTEGER;
    BG : BANGHI;
    TENTEP : STRING[80];
    DTB : REAL;
BEGIN
    CLRSCR;
    N:= 0; TG := 0;
    TENTEP := 'KQ.DAT';
    ASSIGN(F,TENTEP);
    RESET(F);
    WHILE NOT EOF(F) DO
        BEGIN
            READ(F,BG);
            TG := TG +BG.DIEM;
            N := N+1
        END;
    DTB := TG;
    DTB := DTB/N;
    WRITELN(' Diem trung binh cua lop: ', DTB:4:1);
    CLOSE(F)
END.

```

B. BÀI TẬP ÁP DỤNG

I. Bài tập trắc nghiệm

Hãy chọn phương án trả lời đúng nhất.

Phần cơ bản.

16.1. Phát biểu nào dưới đây là đúng?

- A. Tập tin văn bản muốn sử dụng phải định nghĩa bằng từ khoá Type.
- B. Tập tin dạng văn bản chính là tập tin định kiểu với kiểu phần tử là kiểu Char.

- C. Dữ liệu của tập tin định kiểu là bất cứ loại dữ liệu nào.
- D. Tập tin văn bản và tập tin định kiểu có cùng kiểu dữ liệu.

16.2. Tong tập tin dạng văn bản thủ tục Append(biến file) được dùng để?

- A. Kiểm tra kích thước của dữ liệu trong tập tin.
- B. Xoá dòng dữ liệu cuối cùng trong tập tin.
- C. Mở tập tin có tên đã gán cho biến file trong thủ tục assign, cửa sổ file đứng ở vị trí kết thúc của tập tin.
- D. Đóng tập tin có tên đã gán cho biến file trong thủ tục assign, cửa sổ file về đầu file.

16.3. Tong tập tin dạng văn bản phát biểu nào dưới đây là sai?

- A. Thủ tục read(f, biến 1, biến 2, biến n) với biến có kiểu dữ liệu là kiểu số thì các số được ghi trên tập tin lần lượt và mỗi lần đọc xong một số thì cửa sổ file tự động định vị tại vị trí số tiếp theo.
- B. Thủ tục readln(f, biến 1, biến 2, biến n) với biến có kiểu dữ liệu là kiểu số thì các số được ghi trên tập tin lần lượt và mỗi lần đọc xong một số thì cửa sổ file tự động định vị tại vị trí số ở đầu dòng tiếp theo.
- C. Cách đọc của thủ tục read(f, biến 1, biến 2, biến n) và readln(f, biến 1, biến 2, biến n) là như nhau.
- D. Cả 3 đáp án đều sai.

16.4. Giả sử tại thư mục hiện hành có tập tin dạng văn bản với tên là vidu.dat chứa dãy các số nguyên được ghi trên hai dòng như sau:

```

45    -4    30    -22   101
5     24    -23   445

```

Khi chạy chương trình:

```
Uses CRT;
```

```
Var f: Text; n: integer;
```

```
Begin
```

```
Assign(f, 'vidu.dat');
```

```
Reset(f);
```

```
While not Eof(f) do
```

```
Begin
```

```
While not EOLN(f) do
```

```
Begin
```

```
Read(f,n);
```

```
Write(n:5);
```

```
End;
```

```

End;
Close(f);
END.

```

thì sẽ cho kết quả nào?:

- A. -15 -4 30 -22 101
- B. 15 24 -23 445
- C. 15 24 -23 445 -15 -4 30 -22 101
- D. -15 -4 30 -22 101 15 24 -23 445

16.5. Giả sử tại thư mục hiện hành có tập tin dạng văn bản với tên là vidu.dat chứa dãy các số nguyên được ghi trên hai dòng như sau:

```

-15    -4    30    -22    101
15    24    -23    445

```

Khi chạy chương trình:

```
Uses CRT;
```

```
Var f: Text; n: integer;
```

```
Begin
```

```
    Assign(f, 'vidu.dat');
```

```
    Reset(f);
```

```
    While not Eof(f) do
```

```
        Begin Readln(f,n);
```

```
                Write(n:5);
```

```
    End;
```

```
    Close(f);
```

```
END.
```

thì sẽ cho kết quả nào?:

- A. -15 -4
- B. 15 24
- C. -15 15
- D. 15 -15

Phần nâng cao

16.6. Giả sử tại thư mục hiện thời có tập tin dạng văn bản với tên là vid.inp chứa dãy các số được ghi trên hai dòng như sau:

```

5.0000000000E+00    1.123    1.5000000000E+00
1.12321    2.0000000000E+00    15.5

```

Khi chạy chương trình

```
Uses CRT;
```

```
Var f: Text; x: real;
```

```
Begin
```

```

Assign(f, 'vidu.inp');
Reset(f);
While not Eof(f) do
Begin Read(f,x);
      Writeln(x);
End;
Close(f);

```

END.

thì sẽ cho kết quả nào dưới đây?

- A. 5.000000000E+ 01
 1.123000000E+00
 1.500000000E+00
 1.123210000E+00
 2.000000000E+00
 1.550000000E+01
- B. 5.000000000E+00
 1.123000000E+00
 1.500000000E+00
- C. 5.000000000E
 1.123000000E
 1.500000000E
 1.123210000E
 2.000000000E
- D. 5.000000000E-01
 1.123000000E-00
 1.500000000E-00
 1.123210000E-00
 2.000000000E-00
 1.550000000E-01

16.7. Giả sử tại thư mục hiện thời có tập tin dạng văn bản với tên là vidu.inp chứa dãy các số được ghi trên hai dòng như sau:

5.000000000E-00	1.123	1.500000000E+00
1.12321	2.000000000E+00	15.5

Khi chạy chương trình

```

Uses CFT;
Var f: Text; x: real;
Begin

```

```

Assign(f, 'vidu.inp');
Reset(f);
While not Eof(f) do
Begin Readln(f,x);
           Writeln(x:5:3);
End;
Close(f);

```

END.

thì sẽ cho kết quả nào dưới đây?

- A. 5
1.12321
- B. 1.12321
5
- C. 5.000
1.123
- D. 5.0000000
1.12321

16.8. Giả sử tại thư mục hiện hành có tập tin dạng văn bản với tên vanban.txt chứa các dòng văn bản như sau:

```

aaaaabbbbcc
dddeefff
gghhhkkkkk

```

Khi chạy chương trình:

```
Uses CRT;
```

```
Var f: Text; s:string[5]; I: integer;
```

```
Begin
```

```
    Assign(f, 'vanban.txt');
```

```
    Reset(f);
```

```
    For i:=1 to 3 do
```

```
    Begin Readln(f,s);
```

```
           Writeln('**',s);
```

```
    End;
```

```
    Close(f);
```

END.

Thì sẽ cho kết quả nào dưới đây?

- A. **aaaaabbbbcc
**dddeefff

****gghhhkkkkk**

B. **aaaaa**

dddee

gghhh

C. ****aaaaa**

****bbbbbb**

****cc**

D. Tất cả đều sai

16.9. Trong tập tin dạng văn bản hàm EOLN(biến file) có chức năng?

A. Trả về giá trị True nếu cửa sổ file đang ở vị trí cuối file.

B. Trả về giá trị False nếu cửa sổ file đang ở vị trí cuối file.

C. Trả về giá trị True nếu tại thời điểm thực hiện hàm cửa sổ file đang ở vị trí cuối dòng, ngược lại hàm trả về giá trị false.

D. Trả về giá trị False nếu tại thời điểm thực hiện hàm cửa sổ file đang ở vị trí cuối dòng, ngược lại hàm trả về giá trị True.

16.10. Giả sử tại thư mục hiện hành có tập tin dạng văn bản với tên vanban.txt chứa các dòng văn bản như sau:

Aaaaaa bbb cc

Dddee fff

Cho biết sau mỗi dòng đều có cặp ký tự #13 và #10. Khi chạy chương trình:

Uses CRT;

Var f: Text; s:string[5]; l: integer;

Begin

Assign(f, 'vanban.txt');

Append(f);

Write(f, 'AAAA BBB');

Writeln(f, 'CC EEEE');

Writeln(f, 'MMMMM NN PPP');

l:=230; s:= 'TTTTT';

Write(l:5, s:8);

Close(f);

END.

thì sẽ cho kết quả nào dưới đây?

A. **Aaaaaa bbb cc**

Dddee fff

B. **Aaaaaa bbb cc**

Dddee fff

- Aaaaa bbb cc
Dddee fff
- C. Aaaaa bbb cc
Dddee fff
AAAA BBBCC EEEE
MMMMM NN PPP
230 TTTT
- D. Aaaaa bbb cc
Dddee fff
AAAA BBBCC EEEE
MMMMM NN PPP

Phần nâng cao

16.11. Đoạn chương trình dưới đây cho kết quả là gì?

```
Var    ketqua: Text;
      A: Integer;
      B: Real;
      C: String[20];
      D: Boolean;
```

```
Begin
A:= 34; B:= 3.14; C:= 'END.'; D:= true;
Assign(ketqua, 'ketqua.text');
Rewrite(Ketqua);
Write(Ketqua, 'ket qua la:');
Write(Ketqua, A:10);
Write(Ketqua, B:10:4);
Writeln(Ketqua, C);
Writeln(' Dong 2!':10,D );
Close(ketqua);
End.
```

- A. Ket qua la 34 3.14 END
Dong 2! True
- B. 34 3.14 END
Dong 2! True
- C. 34 3.14 END
- D. Dong 2! True.

II. Bài tập tự luận

Phần cơ bản.

16.1. Viết chương trình thực hiện các công việc sau:

- Tạo một tập tin mới có tên là `dulieu.dat` lưu tại thư mục hiện hành sau đó cho lấy ngẫu nhiên các số nguyên dương bé hơn 100, nếu số đó khác không thì ghi vào tập tin và lặp lại, nếu số đó bằng 0 thì dừng.
- Đọc các số đang lưu trong tập tin và ghi lên màn hình.
- Sắp xếp các số lưu trong tập tin theo trật tự không giảm và ghi dãy kết quả ra màn hình.

16.2. Đề quản lí kết quả thi tuyển vào lớp 10 của học sinh phổ thông mỗi thí sinh cần lưu giữ các thông tin sau: Số báo danh, họ và tên, điểm thi các môn toán, lí, hoá. Viết chương trình thực hiện công việc sau:

- Tạo một tập tin mới có tên là `hocsinh.dat` để nhập họ tên thí sinh (chỉ nhập tên) và điểm các môn. Việc nhập được lặp lại cho đến khi họ tên thí sinh là xâu rỗng.
- Sắp xếp theo thứ tự alphabet của danh sách thí sinh đã nhập vào.
- Hiện thị danh sách thí sinh lên màn hình kèm theo điểm các môn và tổng điểm.
- Hiện thị danh sách thí sinh trúng tuyển kèm theo điểm các môn và tổng điểm. Thí sinh trúng tuyển là thí sinh có tổng điểm từ 20 điểm trở lên và không có môn nào có điểm dưới 2.

16.3. Viết chương trình đọc một tập tin dạng văn bản với tên tập tin được nhập từ bàn phím, sau đó thông báo lên màn hình số dòng và số từ có trong tập tin.

16.4. Viết chương trình cho người sử dụng nhập từ bàn phím một dãy số nguyên dương, chừng nào số nhập vào là số 0 thì kết thúc. Dãy số nhập vào được lưu trong một tập tin dạng văn bản, giữa hai số đặt cách nhau ít nhất một kí tự trắng. Tên của tập tin được nhập vào từ bàn phím.

16.5. Giả sử có một tập tin dạng văn bản, lưu dãy các số nguyên dương trên nhiều dòng và mỗi dòng lưu đúng 10 số, giữa hai số cách nhau ít nhất một kí tự trắng (dòng cuối cùng có thể ít hơn 10 số). Viết chương trình đọc dãy số từ tập tin, tên tập tin được nhập vào từ bàn phím, sau đó thông báo lên màn hình trong tập tin có bao nhiêu số chẵn.

16.6. Viết chương trình đọc một tập tin dạng văn bản đã có trên đĩa. Sau đó đổi tất cả các kí tự đầu từ thành in hoa, còn các kí tự khác thành in thường.

16.7. Viết chương trình đọc một tập tin văn bản rồi hiện thị lại văn bản ấy với số thứ tự ở đầu mỗi dòng.

16.8. Viết chương trình đọc một tập tin dạng văn bản rồi hiển thị bảng tần số xuất hiện của mỗi kí tự có trong văn bản (chỉ quan tâm đến các kí tự từ A đến Z không phân biệt chữ hoa, chữ thường).

16.9. Viết chương trình nối hai tập tin cùng kiểu Integer.

16.10. Để quản lí điểm học tập của học sinh phổ thông, mỗi môn học người ta cần quản lí các thông tin sau:

- Điểm kiểm tra miệng có hệ số 1, gồm 1 cột điểm.
- Điểm kiểm tra 15 phút có hệ số 1, gồm 2 cột điểm.
- Điểm kiểm tra 1 tiết có hệ số 2, gồm 2 cột điểm.
- Điểm thi học kì, gồm 1 cột điểm.

Người ta quy định cách tính điểm cuối học kì như sau:

- Điểm trung bình kiểm tra (TBKT) là điểm trung bình của các điểm kiểm tra.

- Điểm trung bình môn (TBM) = $(2 \cdot \text{TBKT} + \text{Điểm thi học kì}) / 2$

Hãy tổ chức kiểu dữ liệu bản ghi tương ứng và viết chương trình thực hiện các công việc sau:

- a. Nhập danh sách học sinh và các cột điểm (Khi nào nhập họ tên học sinh là xâu rỗng thì kết thúc).
- b. Bổ sung thêm họ tên và các cột điểm của 1 học sinh mới.
- c. Sắp xếp lại các bản ghi trong tập tin theo trật tự alphabet của họ tên học sinh.
- d. Hiển thị danh sách học sinh cùng các cột điểm và điểm TBM.
- e. Thông báo lên màn hình danh sách các học sinh thi lại môn này.

Chương 6

CHƯƠNG TRÌNH CON VÀ LẬP TRÌNH CÓ CẤU TRÚC

Khi viết chương trình giải các bài toán phức tạp, chương trình thường rất dài, người đọc rất khó nhận biết được chương trình thực hiện công việc gì. Vấn đề đặt ra là phải cấu trúc chương trình như thế nào để dễ đọc, dễ hiểu. Mặt khác, việc giải quyết các bài toán lớn thường đòi hỏi phải phân thành các bài toán con. Vì vậy, khi lập trình cần phải chia chương trình thành các chương trình con. Có hai loại chương trình con: thủ tục và hàm. Hai loại tham số: tham số giá trị và tham số biến. Hai loại biến: biến toàn cục và biến cục bộ.

Bài 17

CHƯƠNG TRÌNH CON VÀ PHÂN LOẠI

A. PHƯƠNG PHÁP GIẢI TOÁN

1. Khái niệm

Trong khi lập trình chúng ta thường gặp những đoạn chương trình được lặp đi lặp lại nhiều lần ở những chỗ khác nhau. Để tránh rườm rà, những đoạn chương trình này được thay thế bởi các đoạn chương trình con tương ứng và khi cần, ta chỉ việc gọi đoạn chương trình con đó ra mà không phải viết lại cả đoạn chương trình đó.

Đối với những bài toán lớn và phức tạp sẽ tương ứng với một chương trình có thể rất lớn và dài. Do đó việc gỡ rối và hiệu chỉnh sẽ rất khó khăn. Ta có thể phân tách vấn đề phức tạp đó ra thành các vấn đề nhỏ hơn để dễ kiểm tra, gỡ rối từng khối một và sau đó ghép lại thành một chương trình lớn. Mỗi chương trình con có thể phân công cho một lập trình viên hoặc một nhóm cán bộ lập trình đảm nhiệm. Đó là ý tưởng cơ bản của lập trình có cấu trúc.

Chương trình con là một dãy lệnh mô tả một số thao tác nhất định và có thể được thực hiện ở nhiều vị trí trong chương trình.

Lợi ích của việc sử dụng chương trình con:

- + Tránh việc phải viết lặp lại một nhóm lệnh khi nhóm lệnh này được thực hiện nhiều lần khác nhau trong chương trình.
- + Chương trình dễ đọc, dễ hiểu, dễ kiểm tra phát hiện lỗi và sửa sai.
- + Có thể giao cho nhiều người cùng viết một chương trình.
- + Thuận tiện cho việc nâng cấp chương trình.

- + Mở rộng khả năng ngôn ngữ.
- + Phục vụ cho quá trình trừu tượng hoá.

2. Phân loại và cấu trúc chương trình con

Có hai loại chương trình con đó là Hàm và Thủ tục.

- + Hàm (Function) là chương trình con thực hiện một số thao tác nào đó và trả về một giá trị qua tên của nó.
- + Thủ tục (Procedure) là chương trình con thực hiện các thao tác nhất định nhưng không trả về giá trị nào qua tên của nó.
 - Chương trình con đóng vai trò quan trọng trong lập trình, đặc biệt là trong lập trình cấu trúc.
 - Cấu trúc chương trình con và vị trí của nó trong chương trình chính: Chương trình con được viết ở phần khai báo. Chương trình con có phần đầu, phần khai báo và phần thân. Phần khai báo có thể có hoặc không.
 - Chương trình con có thể có tham số hình thức khi khai báo và được thay bằng tham số thực sự khi gọi chương trình con.
 - Chương trình con được gọi bằng tên của nó.
 - Trong chương trình con có khai báo và sử dụng các biến và gọi là biến cục bộ.
 - Một chương trình con có thể có hoặc không có tham số hình thức, có thể có hoặc không có biến cục bộ.

B. BÀI TẬP ÁP DỤNG

Bài tập trắc nghiệm

Hãy chọn phương án trả lời đúng nhất.

Phần cơ bản

17.1. Lập trình có cấu trúc còn gọi là:

- A. Lập trình từ gốc đến ngọn.
- B. Lập trình có phân mảnh.
- C. Lập trình từ trên xuống.
- D. Tất cả các phương án trên.

17.2. Ngôn ngữ lập trình Turbo Pascal cho phép xây dựng mấy loại chương trình con?

- A. 1 loại
- B. 2 loại
- C. 3 loại
- D. 4 loại

17.3. Sự khác nhau cơ bản giữa chương trình con hàm và thủ tục là:

- A. Hàm không khác thủ tục.
- B. Hàm khác thủ tục ở phần khai báo.
- C. Hàm trả về giá trị còn thủ tục thì không.
- D. Đáp án b và c.

17.4. Biến toàn cục là:

- A. Biến khai báo trong chương trình con thủ tục.
- B. Biến khai báo trong chương trình con hàm.
- C. Biến khai báo bất kì ở đâu trong chương trình.
- D. Biến được khai báo sau từ khoá Var của chương trình chính.

17.5. Biến địa phương là:

- A. Biến được khai báo sau từ khoá Var của chương trình con.
- B. Biến được khai báo sau từ khoá Var của chương trình chính.
- C. Biến được khai báo trong chương trình con hàm và thủ tục.
- D. Đáp án a và c đều đúng.

Phần nâng cao

17.6. Hàm có thể nhận kiểu dữ liệu trả về là một trong các kiểu sau:

- A. Real, Integer, Array, Boolean.
- B. Record, Real, Integer, Byte, Boolean.
- C. File, Integer, Array, String, Boolean.
- D. Integer, Real, Char, Boolean, String.

17.7. Tham số hình thức của thủ tục có mấy loại:

- A. 1 loại.
- B. 2 loại.
- C. 3 loại.
- D. Không phân loại.

17.8. Sự khác nhau giữa tham biến và tham trị trong khai báo của chương trình con là:

- A. Tham trị phải định nghĩa bằng từ khoá Type.
- B. Tham trị phải khai báo sau từ khoá Var.
- C. Tham biến phải có từ khoá Var đứng trước.
- D. không khác nhau.

17.9. Phát biểu nào dưới đây là sai ?

- A. Một thủ tục có thể không có tham số hình thức nào.
- B. Một thủ tục có thể có một hoặc nhiều tham số hình thức.
- C. Một thủ tục bắt buộc chỉ có một tham số hình thức.
- D. Đáp án A và B đều đúng.

17.10. Trong lời gọi thủ tục có tham số hình thức thì danh sách các tham số được viết như thế nào?

- A. Danh sách các tham số cách nhau bởi dấu chấm phẩy.
- B. Danh sách các tham số liền nhau.
- C. Danh sách các tham số cách nhau bởi dấu phẩy.
- D. Danh sách các tham số được viết sau từ khoá Va:

II. Bài tập tự luận

- 17.1. Một chương trình như thế nào thì được gọi là chương trình con ?
- 17.2. Tại sao lại sử dụng chương trình con? Lợi ích của nó.
- 17.3. Trình bày cấu trúc của một chương trình, so sánh với cấu trúc của chương trình con.

Bài 18

VÍ DỤ VỀ CÁCH VIẾT VÀ SỬ DỤNG CHƯƠNG TRÌNH CON

A. PHƯƠNG PHÁP GIẢI TOÁN

1. Cấu trúc và vị trí của chương trình con trong chương trình chính.

```
Program  tên_chương_trình_chính;  
Uses  khai báo thư viện sử dụng;  
Const  khai báo hằng;  
Type  khai báo kiểu dữ liệu;  
Var  khai báo biến;  
Procedure  tên_thủ_tục(danh sách các tham số);  
    Các khai báo của thủ tục;  
    Begin  
        Các lệnh của thủ tục;  
    End;  
Function  Tên_hàm(danh sách các t/số hình thức):Kiểu_DL_của_hàm;  
    {Khai báo các biến riêng của hàm (nếu cần)}  
Begin  
    {Thân hàm}  
End;  
BEGIN  
    Các lệnh của chương trình chính;  
    Lời gọi thực hiện hàm và thủ tục;  
END.
```

2. Hai loại chương trình con.

a. Thủ tục (Procedure)

```
Procedure  Tên_thủ_tục(danh sách các tham số hình thức);  
    {Khai báo các biến riêng của thủ tục (nếu cần)}  
Begin  
    {Thân thủ tục}
```

End;

Lời gọi thủ tục:

Tên_thủ_tục(danh sách tham số thực sự);

Chú ý: Các CTC được đặt ở phần khai báo của chương trình chính.

b. Hàm(Function)

Function Tên_hàm(danh sách các t/số hình thức):Kiểu_DL_của_hàm;
{Khai báo các biến riêng của hàm (nếu cần)}

Begin

{Thân hàm}

End;

Lời gọi hàm:

Tên_hàm(danh sách tham số thực sự);

3. Sự khác nhau giữa thủ tục và hàm

- Hàm trả lại giá trị thông qua tên hàm và có thể tham gia vào các biểu thức tính toán.
- Thủ tục không trả lại giá trị do đó nó không thể tham gia vào các biểu thức tính toán.
- Trong thân hàm bao giờ cũng có một dòng lệnh gán cho tên hàm một giá trị.
- Các tham số sau tên hàm và thủ tục được gọi là tham số hình thức hay còn được gọi là các đối số.
- Trong chương trình con các tham số hình thức có hai loại:
 - + Các tham số khai báo không có từ khoá VAR được gọi là tham số giá trị. Khi khai báo theo kiểu tham số giá trị thì chương trình con chỉ được phép dùng giá trị của các tham số này mà không thể thay đổi giá trị của nó ở ngoài chương trình chính. Điều đó có nghĩa là giá trị của tham số trước và sau khi gọi chương trình con đều như nhau.
 - + Các tham số khai báo sau từ khoá VAR được gọi là tham số biến, nghĩa là có thể thay đổi giá trị của các tham số truyền theo kiểu tham biến trong chương trình con và khi ra khỏi chương trình con các biến được truyền vào vẫn giữ nguyên giá trị như đã thay đổi ở trong chương trình con.
- Tham số thực sự là các tham số dùng trong lời gọi hàm hay thủ tục. Danh sách các tham số thực sự trong lời gọi phải tương ứng với danh sách các tham số hình thức trong khai báo chương trình con và chúng phải tương ứng về kiểu.

4. Biến toàn cục, biến cục bộ và cách truyền tham số

a. Khái niệm biến toàn cục, biến cục bộ

Biến toàn cục: Là các biến được khai báo ở phần khai báo của chương trình chính và có phạm vi hoạt động trong toàn bộ chương trình. Có thể sử dụng và làm thay đổi giá trị của biến toàn cục nhờ các lệnh ở trong chương trình chính cũng như các lệnh ở trong chương trình con.

Biến cục bộ: Là các biến khai báo ở phần khai báo của chương trình con và chúng chỉ có tầm hoạt động trong chương trình con đó nên còn được gọi là biến địa phương. Chúng được cấp phát bộ nhớ khi chương trình con được gọi tới và bị xoá khỏi bộ nhớ khi ra khỏi chương trình con.

Nếu tên biến cục bộ của một chương trình con trùng với tên biến toàn cục thì máy cũng không bị nhầm lẫn, máy sẽ dùng hai ô nhớ khác nhau để lưu trữ hai biến và khi ra khỏi chương trình con biến cục bộ sẽ bị xoá khỏi bộ nhớ.

b. Cách truyền tham số cho chương trình con

Khi gặp một lời gọi chương trình con, máy sẽ thực hiện các công việc sau:

- + Cấp phát bộ nhớ cho các đối số, các biến cục bộ.
- + Truyền các giá trị của các tham số thực sự cho các tham trị ở chương trình con theo giá trị tương ứng, các địa chỉ của tham số thực sự cho các tham biến.

- + Thực hiện các lệnh ở trong chương trình con.

Trong khi thực hiện chương trình con, các biến cục bộ và các tham số giá trị có thể bị thay đổi nhưng không ảnh hưởng đến các biến ở bên ngoài. Trái lại mọi thay đổi của tham biến ở trong chương trình con sẽ kéo theo sự thay đổi giá trị của các tham số thực sự tương ứng ở ngoài chương trình chính.

Khi kết thúc chương trình con, máy sẽ giải phóng bộ nhớ của các biến cục bộ và các đối số.

Ví dụ 1: Giải phương trình bậc hai $ax^2 + bx + c = 0$.

```
Program Phuong_trinh_bac_hai;
```

```
Var a,b,c,x1,x2,dt:Real;
```

```
Procedure Nhap;
```

```
Begin
```

```
Write('Nhap cac he so a, b, c:');Readln(a,b,c);
```

```
End;
```

```
Function Delta(a,b,c:Real):Real;
```

```
Begin
```

```
Delta:=b*b-4*a*c;
```

```
End;
```

```
Procedure NghiemPB(delta:Real);
```

```
Begin
```



```

        Writeln('X1 = ',(-b-sqrt(delta))/(2*a):8:2);
        Writeln('X2 = ',(-b+sqrt(delta))/(2*a):8:2);
    End;
Begin
    Nhap;
    dt:=Delta(a,b,c);
    If dt> 0 Then NghiempB(dt)
    Else
        If dt=0 Then
            Writeln('Phuong trinh co nghiep kep X1=X2=',-
b/(2*a):8:2)
            Else Writeln('Phuong trinh vo nghiep');
        Readln;
    End.

```

Ví dụ 2: Tìm USCLN của hai số a và b

```

Program VD2_USCLN;
Var a,b:word;
Function USCLN(a,b:word):word;
Begin
    While a<>b Do If a > b then a:=a-b Else b:=b-a;
    USCLN:=a;
End;
BEGIN
    Write('nhap so a= '); Readln(a);
    Write('nhap so b= '); Readln(b);
    Writeln('USCLN cua ', a, ' va ', b, ' la: ',USCLN(a,b));
    Readln;
END.

```

5. Tính đệ quy của chương trình con

Khái niệm: chương trình con đệ quy là chương trình con tự động gọi đến nó một cách trực tiếp hoặc gián tiếp thông qua một chương trình con khác.

Ưu điểm của chương trình con đệ quy: chương trình con đệ quy dẫn đến chương trình ngắn gọn, lập trình đơn giản nhưng nó có nhược điểm là tốn bộ nhớ để chứa các kết quả trung gian.

Ví dụ: Tính giai thừa qua định nghĩa:

$$n! = 1.2.3...(n-1).n$$

Hoặc định nghĩa

$$n! = \begin{cases} 1 & \text{nếu } n = 0 \\ n.(n-1)! & \text{nếu } n > 0 \end{cases}$$

Khi đó hàm GIAITHUA có thể được định nghĩa như sau :

```
FUNCTION GIAITHUA(n : Integer) : Integer;
```

```
Begin
```

```
  If n=0 then GIAITHUA:=1
```

```
  Else GIAITHUA:=n*GIAITHUA(n-1);
```

```
End;
```

Ví dụ: Bài toán hay trò chơi tháp Hà Nội của các nhà sư. Tương truyền rằng ở một ngôi đền lớn tại Bénarès (Ấn Độ) có một tấm đế bằng đồng trên đó có 3 cái cọc bằng kim cương trong đó Thượng đế khi khai sinh ra thế giới đã xếp 64 cái đĩa bằng vàng trên một cái cọc với đường kính giảm dần. Ngày đêm các nhà sư dịch chuyển đĩa sang một chiếc cọc khác với nguyên tắc sau:

- + Một lần chỉ được dịch chuyển một đĩa.
- + Một đĩa có thể được dịch chuyển từ một cọc này sang cọc khác bất kì.
- + Không được để một đĩa trên một đĩa khác có đường kính nhỏ hơn.

Và ngày tận thế sẽ đến khi 64 chiếc đĩa vàng được dịch chuyển và để làm việc này phải mất trên 58 tỉ năm!

Đánh số 3 cái cọc là C1, C2, C3. Giả sử có n đĩa với đường kính giảm dần và được xếp trên cọc C1. Ta hãy chuyển n đĩa trên cọc C1 sang cọc C3, lấy cọc C2 làm cọc trung gian.

Cách chuyển:

- + Nếu n=1 thì ta chỉ việc chuyển trực tiếp đĩa đó qua cọc C3.
- + Nếu n=2, ta chuyển đĩa 1 từ cọc C1 sang cọc C2, chuyển đĩa 2 từ cọc C1 sang C3, sau đó chuyển đĩa từ cọc C2 sang cọc C3.
- + Tổng quát, khi có n đĩa trên cọc C1, ta chuyển n-1 đĩa trên cọc C1 sang cọc C2, chuyển đĩa thứ n trên cọc C1 sang cọc C3, sau đó chuyển n-1 đĩa trên cọc C2 sang cọc C3.

Ta có tổng quát hoá quá trình chuyển các đĩa của bài toán trên bằng thủ tục sau:

```
Dichuyen(số_đĩa,từ_cọc,tới_cọc,cọc_trung_gian);
```

```
Program Thap_HN;
```

```
Var n : Integer;
```

```
Procedure Dichuyen(n,C1,C2,C3:integer);
```

```
Begin
```

```

    If n=1 Then
        Writeln('C1',C1)
    Else
        Begin
            Dichuyen(n-1,C1,C3,C2);
            Dichuyen(1,C1,C2,C3);
            Dichuyen(n-1,C3,C2,C1);
        End;
    End;
Begin
    Write('Nhap n'); Readln(n);
    Writeln('Qua trinh chuyen cac dia nhu sau:');
    Dichuyen(n,1,2,3);
    Readln;
End.

```

Ví dụ: Dãy số Fibonacci là dãy số $F_1, F_2, F_3 \dots F_n$ được tạo ra với công thức:

$$F_n = F_{n-1} + F_{n-2}$$

Với $F_1 = 1, F_2 = 2$

```

Var n : Integer;
Function Fibonacci(n:Integer):Word;
Begin
    If ((n=1) or (n=2)) Then Fibonacci:=1
    Else Fibonacci := Fibonacci(n-1)+Fibonacci(n-2);
End;
Begin
    Write('Nhap n'); Readln(n);
    Writeln('Ket qua: ',Fibonacci(n));
    Readln;
End.

```

5. Các hàm và thủ tục thường dùng

Clrscr: Xóa màn hình

Gotoxy(x,y:Byte): Di chuyển con trỏ tới vị trí có tọa độ (x,y)

Exit : Kết thúc vô điều kiện CTC hoặc vòng lặp.

Random(n:word): Tạo ra một số nguyên trong phạm vi từ 0 đến n

Int(x:real): Cho ra phần nguyên của x.

Halt(exitcode): Dừng hoạt động của chương trình và quay trở lại hệ điều hành (hoặc màn hình soạn thảo).

B. BÀI TẬP ÁP DỤNG

I. Bài tập trắc nghiệm

Hãy chọn phương án trả lời đúng nhất.

Phần cơ bản.

18.1. Phần khai báo chương trình con được đặt ở đâu trong chương trình chính?

- A. Đặt sau từ khoá Type.
- B. Đặt sau từ khoá Const.
- C. Đặt cuối chương trình chính.
- D. Đặt sau phần khai báo biến.

18.2. Từ khoá để khai báo thủ tục là:

- A. Program.
- B. Procedure.
- C. Function.
- D. Không có câu nào đúng.

18.3. Từ khoá khai báo hàm là:

- A. Function.
- B. Procedure.
- C. Funtion.
- D. Funtionion.

18.4. Phát biểu nào dưới đây là đúng ?

- A. Chương trình con là hàm có thể gọi chương trình con thủ tục.
- B. Chương trình con thủ tục có thể gọi chương trình con hàm.
- C. Một chương trình con có thể có lời gọi các chương trình con khác đã được định nghĩa trước nó.
- D. Cả ba phương án đều đúng.

18.5. Phát biểu nào dưới đây là đúng?

- A. Thời gian tồn tại của biến cục bộ là từ khi chương trình con đó thực hiện cho đến khi kết thúc.
- B. Thời gian tồn tại của biến toàn cục là toàn bộ thời gian chương trình chính hoạt động.
- C. Thời gian hoạt động của biến cục bộ và toàn cục là như nhau.
- D. Đáp án a và b đều đúng.

18.6. Phát biểu nào dưới đây là sai?

- A. Tên của các tham số hình thức và tên của các biến địa phương trong cùng một chương trình con không được trùng nhau.
- B. Tên của các biến địa phương có thể trùng tên với tên biến toàn cục.
- C. Biến toàn cục có thể tham gia trong các chương trình con của nó.
- D. Biến địa phương có thể tham gia trong bất cứ chương trình con nào.

18.7. Trong chương trình khi thực hiện lệnh Tên hàm: = Biểu thức; thì:

- A. Kiểu giá trị của biểu thức có thể nhận bất cứ kiểu gì.
- B. Kiểu của biểu thức phải trùng với kiểu của giá trị trả về của hàm.
- C. Kiểu của biểu thức là kiểu bản ghi.
- D. Kiểu của biểu thức là mảng.

18.8. Phát biểu nào dưới đây là đúng?

- A. Sử dụng hàm như sử dụng thủ tục.
- B. Sử dụng hàm phải thực hiện lời gọi hàm bằng cách: truyền tham số thực tế và được đặt trong một biểu thức có xử lý giá trị của nó.
- C. Sử dụng hàm phải có tham biến, các tham biến cách nhau bởi dấu phẩy.
- D. Sử dụng hàm phải có tham trị.

18.9. Chương trình con đệ quy là:

- A. Chương trình con thủ tục.
- B. Chương trình con hàm.
- C. Chương trình con mà trong thân của nó có ít nhất một câu lệnh là lời gọi đến chính nó.
- D. Chương trình con có tham biến và tham trị.

18.10. Để sử dụng các chương trình con chuẩn của Turbo ta phải sử dụng từ khoá

- A. Unit.
- B. Uses.
- C. System.
- D. Const.

Phần nâng cao

18.11. Đoạn chương trình con dưới đây làm gì?

```
Fuction Giaithua (n: Integer): integer;
```

```
Var i,k: integer;
```

```
Begin
```

```
    I:= 0;
```

```
    K:=1;
```

```
    While i < n do
```

```
    Begin
```

```
        I:= i+1;
```

```
        K:= k*i;
```

```
    End;
```

```
    Giaithua:= k;
```

```
End;
```

- A. Tính giá trị của biểu thức $n(n-1)$.
- B. Tính giá trị của biểu thức $n(n-1)(n-2)$.
- C. Tính giá trị của biểu thức $n(n-1)(n-2)(n-3)$.
- D. Tính giá trị của biểu thức $n!$ (n giai thừa).

18.12. Đoạn chương trình con dưới đây làm gì?

```
Fuction Vidu(x,y :integer) : integer;
```

```
Begin
```

```
    If y = 0 then vidu:= x
```

```
    Else Vidu:= vidu(y, x mod y);
```

```
End;
```

- A. Tìm bội số chung của hai số x và y.

- B. Tính trung bình cộng của hai số x và y.
- C. Tính trung bình nhân của hai số x và y.
- D. Tìm ước số chung lớn nhất của hai số x và y.

18.13. Cho chương trình:

```
Uses CRT;
Var a,b,c: Word;
Procedure    p1(m: word; var n: word);
Begin
    If m > n then
        M:= m -n
    Else m:= n - m;
        N:= 2*m;
End;
Begin
    A:= 5; b:= 8; c:= 10;
    P1(b+c, a);
    Writeln('a=', a, 'b=', b, 'c=',c);
```

End.

Khi chạy chương trình sẽ cho kết quả nào dưới đây?

- | | | | |
|----|-------|-----|------|
| A. | a= 10 | b=8 | c=10 |
| B. | a= 16 | b=6 | c=10 |
| C. | a= 26 | b=8 | c=10 |
| D. | a= 26 | b=8 | c=15 |

18.14. Cho chương trình:

```
Uses CRT;
Var a,b,c: Word;
Procedure    p2(m,n: word);
Var a:word;
Begin
    If m > n then
        a:= m
    Else a:= n;
        Writeln('a=', a, 'b=', b, 'c=',c);
End;
Begin
    A:= 5; b:= 8; c:= 10;
    P2(b,c);
    Writeln('a=', a, 'b=', b, 'c=',c);
```

End.

Khi chạy chương trình sẽ cho kết quả nào dưới đây?

- | | | | |
|----|------|------|------|
| A. | a=10 | b=8 | c=10 |
| | a=5 | b=8 | c=10 |
| B. | a=10 | b=8 | c=10 |
| | a=15 | b=8 | c=10 |
| C. | a=8 | b=10 | c=10 |
| | a=5 | b=8 | c=10 |
| D. | a=10 | b=10 | c=8 |
| | a=5 | b=10 | c=8 |

18.15. Cho chương trình:

Uses CRT;

Var a,b,c: Word;

Procedure p3(m: word; var n: word);

Begin

 If m > n then

 M:= m -n

 Else m:= n - m;

 N:= 2*m;

 A:=m+n;

End;

Begin

 A:= 5; b:= 8; c:= 10;

 P3(b+c, a);

 Writeln('a=', a, 'b=', b, 'c=',c);

End.

Khi chạy chương trình sẽ cho kết quả nào dưới đây?

- | | | | |
|----|------|------|------|
| A. | a=30 | b=10 | c=8 |
| B. | a=39 | b=8 | c=10 |
| C. | a=30 | b=8 | c=10 |
| D. | a=39 | b=10 | c=8 |

18.16. Cho chương trình con

Function S(n:word): word;

Begin

 If n=1 then s:=1

 Else s:= s(n-1)+n;

End;

Chương trình con đã cho ở trên làm gì?

- A. Tính tổng các số tự nhiên.

- B. Tính tổng các số tự nhiên chẵn
- C. Tính tổng các số theo công thức $S=1+2+..+n$.
- D. Tính tổng các số theo công thức $S=1+2+..+n*n$.

18.17. Cho chương trình con

```

Procedure      vidu(n: word);
  Begin
    If n div 2 < > 0 then
      Vidu(n div 2);
      Write(n mod 2);
  End;

```

Chương trình con đã cho ở trên làm gì?

- A. Tìm các số nhị phân của số nguyên dương n cho trước.
- B. In ra màn hình số dạng nhị phân của một số nguyên dương n cho trước.
- C. In ra màn hình các số nhị phân nhỏ hơn 100.
- D. Chuyển một số nguyên dương n thành các số nhị phân.

18.18. Chương trình con dưới đây làm gì?

```

Procedure      baitap(var s: String);
  Var i: integer;
  Begin
    For i:= 1 to length(s) do
      s[i]:= Uppcase(s[i]);
  End;

```

- A. Chuyển chuỗi kí tự hoa qua kí tự thường.
- B. Tính chiều dài của chuỗi s.
- C. Chuyển chuỗi kí tự thường qua chuỗi kí tự hoa.
- D. In chuỗi theo chiều ngược lại.

II. Bài tập tự luận

Phần cơ bản

- 18.1.** Xây dựng hàm $\log(a, b: \text{real})$, trong đó a và b là các số nguyên dương, giá trị trả về của hàm là giá trị của $\log_a b$.
- 18.2.** Xây dựng hàm $\max(a, b: \text{real})$ có giá trị trả về của hàm là giá trị lớn nhất trong hai số a và b.
- 18.3.** Xây dựng hàm $\text{BCNN}(a, b: \text{Word})$ có giá trị trả về của hàm là bội số chung nhỏ nhất của hai số nguyên dương a và b.

- 18.4.** Xây dựng hàm Daonguoc(n: word) có giá trị trả về của hàm là số nguyên có được bằng cách lấy đảo ngược các chữ số của n. Ví dụ khi đảo ngược số 12305 thì có kết quả 50321.
- 18.5.** Xây dựng hàm SNT(n: word) có giá trị trả về của hàm là True nếu n là số nguyên tố và ngược lại thì giá trị trả về của hàm là False.
- 18.6.** Hãy sử dụng hàm SNT trên bài 18.5 để viết thủ tục liệt kê ra màn hình tất cả các số nguyên tố bé hơn hoặc bằng n, mỗi dòng có 10 số, ngoại trừ dòng cuối cùng, giữa hai số cách nhau ít nhất một kí tự trắng. Trong đó n là số nguyên dương cho trước.
- 18.7.** Xây dựng hàm S(n: word), bằng hai cách đệ quy và không đệ quy, có giá trị trả về của hàm là giá trị của tổng $S = 1^2 + 2^2 + 3^2 + \dots + n^2$.
- 18.8.** Xây dựng hàm S(n: word; x: real), bằng hai cách đệ quy và không đệ quy, có giá trị trả về của hàm là giá trị của tổng $S = x \sin x + x^2 \sin^2 x + \dots + x^n \sin^n x$.

Phần nâng cao

- 18.9.** Xây dựng thủ tục để xoá các kí tự trắng dư thừa trong một chuỗi cho trước (kí tự trắng ở vị trí đầu chuỗi, cuối chuỗi và hai kí tự trắng liền nhau ở giữa thì xoá còn lại một).
- 18.10.** Xây dựng thủ tục để đổi tất cả các kí tự thành in hoa của một xâu kí tự cho trước.
- 18.11.** Xây dựng thủ tục để đổi tất cả các kí tự đầu từ thành in hoa, các kí tự khác thành in thường của một xâu kí tự cho trước.
- 18.12.** Xây dựng hàm đếm số từ có trong một xâu kí tự cho trước.
- 18.13.** Xây dựng hàm đảo ngược từ cuối cùng thành từ đầu tiên của một xâu kí tự cho trước. Ví dụ 'Nguyen Van Anh' đảo thành Anh Van Nguyen.
- 18.14.** Cho hai xâu dạng số nguyên s1 và s2 (giả sử có độ dài tương đối lớn mà ta không thể dùng thủ tục Val để đổi qua kiểu số được).
- Xây dựng hàm để tính tổng (theo phép tính số học) của hai xâu dạng số trên. Ví dụ a1 = '898765' và s2 = '888' thì tổng của chúng là '899653'.
 - Xây dựng hàm tính tích của hai xâu s1 và s2.
- 18.15.** Xây dựng thủ tục Tim(var s:string; s1, s2: string) để thực hiện công việc tìm trong xâu s, chỗ nào có xâu con s1 thì thay thế bởi xâu con s2.
- 18.16.** Xây dựng thủ tục Nhapxausonguyen(Var s:string) để nhập xâu biểu diễn số nguyên cho tham biến s. Với yêu cầu: Nếu xâu dạng số s không hợp lệ thì yêu cầu nhập lại.

18.17. Xây dựng thủ tục Nhapxausothuc(Var s:string) để nhập xâu biểu diễn số thực cho tham biến s. Với yêu cầu: Nếu xâu dạng số s không hợp lệ thì yêu cầu nhập lại.

18.18. Để quản lí điểm học tập của học sinh phổ thông, mỗi môn học người ta quản lí các thông tin sau:

- Điểm kiểm tra miệng có hệ số 1, gồm 1 cột điểm.
- Điểm kiểm tra 15 phút có hệ số 1, gồm 2 cột điểm.
- Điểm kiểm tra 1 tiết có hệ số 2, gồm 2 cột điểm.
- Điểm kiểm tra học kì gồm 1 cột điểm.
- Điểm trung bình kiểm tra (TBKT) là điểm trung bình của các điểm kiểm tra.
- Điểm trung bình môn được tính như sau: $TBM = (TBKT * 2 + \text{điểm học kì}) / 2$.

Viết chương trình dưới dạng chương trình con nhập danh sách học sinh và các cột điểm tương ứng. Thông báo lên màn hình danh sách học sinh phải thi lại môn học này (học sinh thi lại là học sinh có điểm trung bình dưới 5).

BÀI 19

THƯ VIỆN CHƯƠNG TRÌNH CON CHUẨN

A. PHƯƠNG PHÁP GIẢI TOÁN

1. Thư viện CRT

- Thư viện CRT chứa các thủ tục liên quan đến việc quản lí và khai thác màn hình và bàn phím.
- Thủ tục Clrscr: xoá màn hình.
- Thủ tục Ttextcolor(c): Đặt màu cho chữ trên màn hình, trong đó c là hằng hoặc biến có giá trị nguyên không âm để xác định màu.
- Thủ tục Textbackground(c): Đặt màu cho nền của màn hình
- Thủ tục Gotoxy(x,y): Đưa con trỏ đến vị trí cột x dòng y của màn hình văn bản.

2. Thư viện Graph của ngôn ngữ lập trình Pascal.

- Thư viện Graph chứa các chương trình con phục vụ khai thác khả năng đồ hoạ của máy tính ở mức độ thông dụng như vẽ điểm, đường, tô màu...
- Các thiết bị và chương trình hỗ trợ đồ hoạ:
 - + Có hai chế độ màn hình: Đồ hoạ và văn bản.
 - + Bảng mạch điều khiển màn hình là thiết bị đảm bảo tương tác giữa bộ xử lí và màn hình để thể hiện các chế độ phân giải và màu sắc

+ Turbo Pascal cung cấp các chương trình điều khiển (có phần mở rộng là BGI) tương ứng với các loại card đồ họa. Khi khởi động chế độ đồ họa cần chỉ ra đường dẫn đến chương trình này.

+ Toạ độ màn hình đồ họa được đánh số từ 0. Cột được đánh số từ phải sang trái, dòng được đánh số từ trên xuống dưới. Giá trị lớn nhất của toạ độ dòng và toạ độ cột được gọi là độ phân giải của màn hình.

+ Để thực hiện được chức năng đồ họa cần sử dụng các thủ tục và hàm trong thư viện Graph.

- Khởi động chế độ đồ họa: `Initgraph(dr,md:integer;pth:string);`

dr: là số hiệu của trình điều khiển BGI.

Md: là số hiệu của độ phân giải.

Pth: là đường dẫn đến các tệp BGI.

- Kết thúc chế độ đồ họa trở về chế độ văn bản: `Closegraph;`

3. Tìm hiểu các thủ tục vẽ điểm, đường và các hình cơ bản.

- Vẽ điểm: `Putpixel(x,y:integer;color:word);`

- Vẽ đường thẳng: `Line(x1,y1,x2,y2:integer);`

`Lineto(x,y:integer);`

`Linerel(dx,dy:integer);`

- Vẽ hình tròn: `Circle(x,y:integer;r:word);`

- Vẽ hình elip: `Ellipse(x,y:integer; stangle,endangle,xr,yr:word);`

- Vẽ hình chữ nhật: `Rectangle(x1,y1,x2,y2:integer);`

- Đặt màu cho nét vẽ: `Setcolor(word);`

Tìm hiểu một số thư viện khác:

- Thư viện System chứa các hàm sơ cấp và các thủ tục vào/ra mà các chương trình đều dùng tới.

- Thư viện Dos chứa các thủ tục cho phép thực hiện trực tiếp các lệnh như tạo thư mục, thiết lập ngày, giờ hệ thống.

- Thư viện Printer cung cấp các thủ tục làm việc với máy in.

II. BÀI TẬP ÁP DỤNG

Bài tập tự luận

19.1. Viết chương trình vẽ một đường thẳng có toạ độ bất kì và có màu đỏ.

19.2. Viết chương trình vẽ một hình chữ nhật có toạ độ bất kì có màu xanh.

19.3. Viết chương trình vẽ một hình tròn có tâm và bán kính nhập vào từ bàn phím và có màu vàng.

19.4. Viết chương trình vẽ một hình elip có tâm và bán kính nhập vào từ bàn phím và có màu tím.

ĐÁP ÁN VÀ HƯỚNG DẪN GIẢI

Chương 1: MỘT SỐ KHÁI NIỆM VỀ LẬP TRÌNH VÀ NGÔN NGỮ LẬP TRÌNH.

Bài 1: Khái niệm lập trình và ngôn ngữ lập trình

Bài tập trắc nghiệm

- | | | |
|---------|---------|----------|
| 1.1. A. | 1.5. A. | 1.9. B. |
| 1.2. C. | 1.6. D. | 1.10. C. |
| 1.3. D. | 1.7. C. | 1.11. D. |
| 1.4. B. | 1.8. D. | 1.12. A. |

Bài 2: Các thành phần của ngôn ngữ lập trình

I. Bài tập trắc nghiệm

- | | | |
|---------|---------|----------|
| 2.1. C. | 2.5. B. | 2.9. B. |
| 2.2. D. | 2.6. D. | 2.10. D. |
| 2.3. A. | 2.7. B. | |
| 2.4. B. | 2.8. D. | |

Phần tự luận Dựa vào sách giáo khoa để trả lời các câu hỏi.

Chương 2: CHƯƠNG TRÌNH ĐƠN GIẢN

Bài 3. Cấu trúc chương trình

Bài tập trắc nghiệm

- | | | |
|---------|---------|----------|
| 3.1. C. | 3.5. C. | 3.9. B. |
| 3.2. B. | 3.6. C. | 3.10. A. |
| 3.3. C. | 3.7. C. | 3.11. C. |
| 3.4. B. | 3.8. A. | |

II. Phần tự luận.

- 3.1. Khai báo tên chương trình sử dụng từ khoá Program.
- 3.2. Khai báo thư viện sử dụng từ khoá Uses.
- 3.3. Khai báo hằng sử dụng từ khoá Const.
- 3.4. Kết hợp các câu 3.1 đến 3.3 để thực hiện.
- 3.5. Sử dụng thêm câu lệnh Write để viết câu "Turbo Pascal".

Bài 4. Một số kiểu dữ liệu chuẩn.

I. Bài tập trắc nghiệm

- | | | | | | | |
|---------|---------|---------|---------|---------|---------|---------|
| 4.1. C. | 4.2. C. | 4.3. B. | 4.4. D. | 4.5. A. | 4.6. B. | 4.7. A. |
|---------|---------|---------|---------|---------|---------|---------|

4.8. D. 4.9. C. 4.10. D. 4.11. C. 4.12. B. 4.13. A. 4.14. C.
4.15. A. 4.16. D. 4.17. B. 4.18. B. 4.19. D. 4.20. A. 4.21. D.

II. Phần tự luận

- Dựa vào sách giáo khoa để trả lời các câu hỏi.

Bài 5: Khai báo biến

I. Bài tập trắc nghiệm

5.1. B. 5.3. C. 5.5. B. 5.7. C. 5.9. D.
5.2. C. 5.4. B. 5.6. A. 5.8. A. 5.10. B.

II. Phần tự luận

- 5.1. Xem chức năng của hàm chr và ord.
- 5.2. Kí tự thường hơn kí tự hoa 32 giá trị trong bảng mã ASCII.
- 5.3. Tính giá trị của biểu thức, dựa vào kết quả để xác nhận kiểu dữ liệu.
- 5.4. Dựa vào bảng mã ASCII để so sánh.
- 5.5. Dựa vào kiểu dữ liệu của các biến và kích thước của mỗi kiểu dữ liệu để tính tổng bộ nhớ phải sử dụng.

Bài 6: Phép toán, biểu thức, câu lệnh gán.

I. Bài tập trắc nghiệm

6.1. C. 6.2. D. 6.3. D. 6.4. B. 6.5. B.
6.6. C. 6.7. C. 6.8. B. 6.9. B. 6.10. D.
6.11. A. 6.12. A. 6.13. D. 6.14. C. 6.15. D.

II. Phần tự luận

- 6.1. Khai báo các biến làm nghiệm và giá trị delta có kiểu thực.
- 6.2. Khai báo biến có kiểu thực
- 6.3. Sử dụng cú pháp khai báo biến để thực hiện.
- 6.4. Chú ý tính tương thích của biến với giá trị gán vào.
- 6.5. Xem lại các phép toán trên các kiểu dữ liệu.

Bài 7: Các thủ tục chuẩn vào/ra đơn giản.

I. Bài tập trắc nghiệm

7.1. C. 7.2. D. 7.3. D. 7.4. C. 7.5. B.
7.6. D. 7.7. D. 7.8. C. 7.9. B. 7.10. A.
7.11. C. 7.12. D. 7.13. B. 7.14. B. 7.15. D.

II. Phần tự luận

- 7.1 đến 7.5: Xem lí thuyết ở sách giáo khoa để giải quyết.
- 7.6. Hướng dẫn:

- Khai báo các biến kiểu thực.
- Nhập giá trị của đáy ngắn, đáy dài, chiều cao.
- Tính diện tích hình thang theo công thức $S = (a+b)h/2$.
- Hiển thị kết quả ra màn hình.

7.7. Hướng dẫn:

- Tính giá trị của các biểu thức
- Dựa vào kết quả để xác định kiểu dữ liệu.

7.8. Hướng dẫn:

- Khai báo dữ liệu.
- Nhập các giá trị vào.
- Tính tiền theo công thức.
- Hiển thị kết quả ra màn hình.

7.9. Hướng dẫn:

- Khai báo dữ liệu chứa tọa độ các đỉnh.
- Nhập các giá trị vào.
- Tính diện tích và chu vi theo công thức.
- Hiển thị kết quả ra màn hình.

7.10. Hướng dẫn:

- Khai báo dữ liệu chứa độ dài ba cạnh.
- Nhập các giá trị vào.
- Tính diện tích và chu vi, độ dài ba đường cao, ba đường phân giác theo công thức.
- Hiển thị kết quả ra màn hình.

7.11. Hướng dẫn:

- Khai báo dữ liệu chứa độ dài bán kính hình cầu.
- Nhập các giá trị vào.
- Tính diện tích và thể tích theo công thức.
- Hiển thị kết quả ra màn hình.

7.12. Hướng dẫn:

- Khai báo dữ liệu.
- Nhập các giá trị vào.
- Tính chu vi theo công thức.
- Hiển thị kết quả ra màn hình.

7.13. Dựa vào quy luật của cấp số cộng để tính.

Bài 8. Soạn thảo, dịch, thực hiện và hiệu chỉnh chương trình.

I. Bài tập trắc nghiệm

8.1.D. 8.3. C. 8.5. C. 8.7. A. 8.9. C.
8.2. B. 8.4. C. 8.6. B. 8.8. B. 8.10. B.

II. Phần tự luận

- 8.1. Chú ý các bước cơ bản trong soạn thảo.
- 8.2. Thực hiện thao tác lưu dữ liệu một cách thường xuyên.
- 8.3. Xem một số tổ hợp phím tắt.
- 8.4. Tập thực hành và rút kinh nghiệm.
- 8.5. Xem chức năng các Menu.

Chương 3. CẤU TRÚC RỄ NHÁNH VÀ LẬP

Bài 9. Cấu trúc rẽ nhánh

I. Bài tập trắc nghiệm

9.1. C. 9.2. D. 9.3. D. 9.4. A. 9.5. B. 9.6. C. 9.7. A.
9.8. B. 9.9. D. 9.10. C. 9.11. B. 9.12. C. 9.13. B. 9.14. C.
9.15. A. 9.16. B.

II. Phần tự luận

9.1. Hướng dẫn:

- Khai báo biến a,b kiểu thực
- Nhập giá trị cho a và b
- Biện luận nghiệm theo a và b.
- Kiểm tra nếu $a > 0$ thì thông báo nghiệm của phương trình là $-b/a$
- Kiểm tra nếu $a = 0$ và $b = 0$ thì phương trình có vô số nghiệm
- Nếu $a = 0$ và b khác 0 thì phương trình vô nghiệm.
- Thông báo kết quả ra màn hình.

9.2. Hướng dẫn:

- Khai báo biến a,b,c kiểu thực.
- Nhập giá trị cho a, b, c.
- Biện luận nghiệm theo delta, trong đó $\Delta = b^2 - 4ac$
- Kiểm tra nếu $\Delta > 0$ phương trình có hai nghiệm phân biệt.
- Kiểm tra nếu $\Delta < 0$ phương trình vô nghiệm.
- Kiểm tra nếu $\Delta = 0$ phương trình có một nghiệm kép.
- Thông báo kết quả ra màn hình.

9.3. Hướng dẫn:

- Khai báo hai số a và b, khai báo hai biến max và min chứa giá trị lớn nhất và bé nhất cùng kiểu với a và b.
- Kiểm tra và gán các giá trị bé nhất cho min và lớn nhất cho max

- Hiện thị kết quả.

9.4. Hướng dẫn:

- Khai báo ba số a, b, c ; khai báo hai biến $\max$ và $\min$ chứa giá trị lớn nhất và bé nhất cùng kiểu với a, b, c .

- Kiểm tra và gán các giá trị bé nhất cho $\min$ và lớn nhất cho $\max$

- Hiện thị kết quả.

9.5. Hướng dẫn:

- Khai báo 3 biến kiểu nguyên.

- Nhập giá trị của 3 biến làm độ dài 3 cạnh tam giác, chú ý phải kiểm tra các giá trị nguyên dương.

- Kiểm tra điều kiện để các số là 3 cạnh của 1 tam giác.

- Thông báo kết quả ra màn hình.

9.6. Hướng dẫn:

- Khai báo các biến của hệ.

- Nhập giá trị cho các biến.

- Biện luận theo từng trường hợp

- In ra kết quả.

9.7. Hướng dẫn:

- Khai báo một biến để nhập điểm có kiểu số nguyên.

- Nhập giá trị, kiểm tra giá trị nhập vào thoả mãn điều kiện lớn hơn hoặc bằng 0 và nhỏ hơn hoặc bằng 10.

9.8. Hướng dẫn:

- Quy về phương trình bậc hai

- Xem hướng dẫn bài 9.2.

9.9. Hướng dẫn:

- Khai báo 3 biến kiểu nguyên.

- Nhập giá trị và kiểm tra điều kiện các số đó phải dương.

- Kiểm tra các số, so sánh giá trị và in ra màn hình theo trật tự tăng dần.

9.10. Hướng dẫn:

- Khai báo biến

- Nhập giá trị

- Tính chu vi và diện tích theo công thức.

- Hiện thị kết quả ra màn hình.

9.11. Hướng dẫn:

- Khai báo hai số nguyên m và n .

- Nhập dữ liệu, kiểm tra hai số phải nguyên dương thì mới xử lí, nếu không thoả mãn nhập lại.

- Kiểm tra nếu m khác n thì xử lí, nếu $m > n$ thì $m := m - n$, ngược lại $n := n - m$.
Tìm được ước chung lớn nhất.
- Tìm được bội chung nhỏ nhất.

9.12. Hướng dẫn:

- Khai báo dữ liệu.
- Nhập số KW tiêu thụ.
- Dựa vào giá trị để tính thành tiền phải trả.
- In kết quả ra màn hình.

Bài 10. Cấu trúc lặp

I. Câu hỏi trắc nghiệm

- 10.1. B. 10.2. C. 10.3. D. 10.4. C. 10.5. D. 10.6. D. 10.7. B.
 10.8. A. 10.9. D. 10.10. C. 10.11. A. 10.12. C. 10.13. D. 10.14. A.
 10.15. B. 10.16. C. 10.17. B. 10.18. C. 10.19. A. 10.20. C. 10.21. A.
 10.22. C. 10.23. A. 10.24. D. 10.25. D. 10.26. A. 10.27. B. 10.28. C.
 10.29. B. 10.30. C. 10.31. A. 10.32. A. 10.33. B.

II. Phân tự luận

10.1. Hướng dẫn:

- Khai báo biến n, s, i .
- Nhập giá trị n .
- Duyệt vòng for và tính tổng theo công thức.
- Hiện thị kết quả.

10.2. Hướng dẫn:

- Khai báo biến s, i .
- Duyệt vòng for và kiểm tra nếu giá trị chẵn ($\text{mod } 2 = 0$) thì tính tổng theo công thức cộng dồn.
- Hiện thị kết quả.

10.3. Hướng dẫn:

- Khai báo biến s, i .
- Duyệt vòng for và kiểm tra nếu giá trị lẻ ($\text{mod } 2 \neq 0$) thì tính tổng theo công thức cộng dồn.
- Hiện thị kết quả.

10.4. Hướng dẫn:

- Sử dụng thuật toán Oclit tìm UCLN.
- Bội chung nhỏ nhất bằng tích hai số chia cho UCLN

10.5. Hướng dẫn:

- Khai báo dữ liệu.

- Nhập vào một số nguyên dương.
- Dùng vòng lặp không xác định để thực hiện công việc sau cho đến khi $n = 0$ thì dừng:
 - + Ghi lên màn hình chữ số cuối cùng của n .
 - + Loại bỏ chữ số cuối cùng bằng cách gán lại $n := n \text{ div } 10$.

10.6. Hướng dẫn:

- Khai báo dữ liệu.
- Nhập hai số m, n nguyên dương ($m < n$), phải kiểm tra điều kiện.
- Dùng lệnh for duyệt tất cả các số từ m đến n , kiểm tra nếu đúng là số hoàn hảo thì thông báo, nếu không thì bỏ qua.

10.7. Hướng dẫn:

- Khai báo dữ liệu.
- Nhập giá trị số n .
- Khởi gán giá trị $i := 2$;
- Sử dụng vòng lặp While lặp lại: chừng nào $i \leq \text{round}(\sqrt{n})$ và $n \bmod i \neq 0$ thì tăng giá trị i lên 1; Sau đó kiểm tra nếu đồng thời $i \neq 1$ và $i > \text{round}(\sqrt{n}) * i * n$ thì n là số nguyên tố; ngược lại thì n không là số nguyên tố.

10.8. Hướng dẫn:

- Khai báo dữ liệu.
- Nhập giá trị n .
- Dùng vòng lặp for duyệt tất cả các giá trị từ 1 đến n và kiểm tra nếu đúng là số nguyên tố thì thông báo, nếu không thì bỏ qua.

10.9. Hướng dẫn:

- Khai báo dữ liệu.
- Khởi gán $a := 1; b := 1$;
- Dùng vòng lặp Repeat lặp lại:
 - $C := a + b$;
 - $A := b; b := c$;
 - Cho đến khi $c > 2000$ thì dừng lại.

- Thông báo kết quả là a .

10.10. Hướng dẫn:

- Khai báo dữ liệu.
- Nhập giá trị n .
- Khởi gán $i := 2$;
- Dùng vòng lặp While để xử lý và in ra màn hình.
 - While $i < n$ do

If $n \bmod i = 0$ then

Begin

If $i < n$ then write(i , ' *')

Else write(i);

$N := n \div i$;

End

else $i := i + 1$;

10.11. Hướng dẫn:

- Khai báo biến g để chứa số gà.
- Sử dụng vòng for duyệt hết tất cả các khả năng của gà từ 1 đến 35. Kiểm tra nếu $2 * g * 4$ ($36 - g$) = 100 thì thông báo số gà là g và số chó là $36 - g$.

10.12. Hướng dẫn:

- Khai báo st , t , k , max .
- Nhập t , k , max .
- Khởi gán biến số tháng $st = 0$;
- Dùng vòng lặp Repeat lặp lại:
 $St := st + 1$;
 $T := t + t * k$;

Cho đến khi $t \geq max$ thì dừng.

- Thông báo kết quả là st .

10.13. Hướng dẫn:

- Khai báo biến
- Duyệt vòng lặp for từ 0 đến 99 và in ra, chú ý có 10 số trên 1 dòng và có 10 dòng.

10.14. Hướng dẫn:

- Khai báo dữ liệu
- Nhập n .
- Kiểm tra n và tính theo công thức.
- Hiện thị kết quả.

10.15. Hướng dẫn:

- Khai báo biến chạy kiểu char.
- Duyệt vòng for và hiện thị.

10.16. Hướng dẫn:

- Áp dụng công thức Taylor với số hạng thứ k của chuỗi $T(k)$ theo công thức truy hồi như sau: $T(k) = T(k-1) * x / k$

10.17. Hướng dẫn:

- Áp dụng công thức Taylor với số hạng thứ k của chuỗi $T(k)$.

10.18. Hướng dẫn:

- Khai báo biến chạy i.
- Duyệt vòng for từ 1 đến 255 và tương ứng với mỗi giá trị in ra mã của nó theo hàm Chr(i).
- Hiển thị 15 giá trị trên một dòng.

10.19. Hướng dẫn:

- Khai báo dữ liệu.
- Nhập giá trị n, khởi tạo các biến ban đầu.
- Dùng cú pháp lặp Repeat và While để kiểm tra và tính.
- Thông báo kết quả ra màn hình.

10.20. Hướng dẫn:

- Khai báo dữ liệu.
- Sử dụng vòng lặp for duyệt 3 số a, b, c từ 1 đến 9 và kiểm tra theo điều kiện $abc = 100a + 10b + c = a^3 + b^3 + c^3$ nếu thoả mãn thì in ra màn hình.

Chương 4. KIỂU DỮ LIỆU CÓ CẤU TRÚC

Bài 11. Kiểu mảng

I. Bài tập trắc nghiệm

- 11.1. D. 11.2. C. 11.3. B. 11.4. C. 11.5. B. 11.6. D. 11.7. A.
 11.8. B. 11.9. C. 11.10. C. 11.11. D. 11.12. C. 11.13. A. 11.14. D.
 11.15. D. 11.16. B. 11.17. C. 11.18. C. 11.19. B. 11.20. D. 11.21. D.
 11.22. A. 11.23. B. 11.24. D. 11.25. B. 11.26. A. 11.27. A. 11.28. C.
 11.29. C. 11.30. B. 11.31. C. 11.32. C. 11.33. D. 11.34. C. 11.35. C.
 11.36. B. 11.37. D. 11.38. A.

II. Bài tập tự luận

11.1. Hướng dẫn:

- Khai báo một mảng chứa các số nguyên.
- Nhập phần tử vào cho mảng.
- Duyệt các phần tử của mảng và in ra màn hình.

11.2. Hướng dẫn:

- Khai báo một mảng chứa các số nguyên.
- Nhập phần tử vào cho mảng.
- a. Duyệt mảng và cộng dồn các giá trị vào tổng, tổng ban đầu có giá trị bằng 0.
- b. Duyệt mảng, kiểm tra nếu giá trị của mảng là dương thì tăng biến đếm lên, sau đó cộng giá trị đó vào tổng, ban đầu biến đếm và tổng nhận giá trị bằng 0.

- c. Duyệt mảng, kiểm tra nếu giá trị của mảng là âm thì tăng biến đếm lên sau đó cộng giá trị đó vào tổng, ban đầu biến đếm và tổng nhận giá trị bằng 0.
- d. Duyệt mảng và cộng dồn các giá trị vào tổng, tổng ban đầu có giá trị bằng 0. Sau khi tính được tổng thì giá trị trung bình bằng tổng chia cho số phần tử của mảng.

1.3. Hướng dẫn:

- Khai báo một mảng chứa các số thực.
- Nhập phần tử vào cho mảng.
- a. Duyệt mảng bằng vòng lặp while, kiểm tra xem các phần tử của mảng có dương hay không, nếu có phần tử thoả mãn thì in ra giá trị và chỉ số của phần tử đó, dừng việc kiểm tra lại. Nếu duyệt hết mảng mà không tìm thấy thì thông báo là không có.
- b. Duyệt mảng bằng vòng lặp while, kiểm tra xem các phần tử của mảng có âm hay không, nếu có phần tử thoả mãn thì in ra giá trị và chỉ số của phần tử đó, dừng việc kiểm tra lại. Nếu duyệt hết mảng mà không tìm thấy thì thông báo là không có.
- c. Duyệt mảng bằng vòng lặp while, kiểm tra xem các phần tử của mảng có dương hay không, nếu có phần tử thoả mãn thì in ra giá trị và chỉ số của phần tử đó và dừng việc kiểm tra lại. Nếu duyệt hết mảng mà không tìm thấy thì thông báo là không có. Lưu ý là duyệt mảng từ sau tới trước.
- d. Duyệt mảng bằng vòng lặp while, kiểm tra xem các phần tử của mảng có âm hay không, nếu có phần tử thoả mãn thì in ra giá trị và chỉ số của phần tử đó và dừng việc kiểm tra lại. Nếu duyệt hết mảng mà không tìm thấy thì thông báo là không có. Lưu ý là duyệt mảng từ sau tới trước.

11.4. Hướng dẫn:

- Khai báo một mảng chứa các số thực.
- Nhập phần tử vào cho mảng.
- a. Duyệt mảng, gán giá trị lớn nhất (max) bằng phần tử đầu tiên rồi lần lượt so sánh các giá trị để tìm ra giá trị lớn nhất, sau khi tìm được thì in ra chỉ số của nó.
- b. Duyệt mảng, gán giá trị bé nhất (min) bằng phần tử đầu tiên rồi lần lượt so sánh các giá trị để tìm ra giá trị bé nhất, sau khi tìm được thì in ra chỉ số của nó.

- c. Dùng vòng lặp không xác định (while) để duyệt mảng, lần lượt so sánh các giá trị âm để tìm ra giá trị âm lớn nhất, sau khi tìm được thì in ra chỉ số của nó.
- d. Duyệt mảng tương tự câu (c), lần lượt so sánh các giá trị dương để tìm ra giá trị dương lớn nhất, sau khi tìm được thì in ra chỉ số của nó.

11.5. Hướng dẫn:

- Khai báo một mảng chứa các số thực.
- Nhập phần tử vào cho mảng.
- Duyệt mảng và so sánh các phần tử theo giải thuật sau:

For i := 1 to n-1 do

For j := i+1 to n do

If a[i] > a[j] then

Begin

Tg:= a[i];

a[i]:= a[j];

a[j]:=tg;

End;

- Hiện thị các giá trị sau khi đã sắp xếp ra màn hình.

11.6. Hướng dẫn:

- Khai báo một mảng chứa các số thực.
- Nhập phần tử vào cho mảng.
- Duyệt mảng và so sánh các phần tử theo giải thuật bài 11.5 nhưng kiểm tra ngược lại.
- Hiện thị các giá trị sau khi đã sắp xếp ra màn hình.

11.7. Hướng dẫn:

- Khai báo một mảng chứa các số nguyên.
- Nhập phần tử vào cho mảng.
- a. Duyệt mảng, sử dụng hàm mod để kiểm tra các số chẵn (chia hết cho 2) nếu thoả mãn thì tính tổng theo cách cộng dồn, ban đầu tổng có giá trị bằng 0.
- b. Duyệt mảng, sử dụng hàm div để kiểm tra các số lẻ nếu thoả mãn thì tính tổng theo cách cộng dồn, ban đầu tổng có giá trị bằng 0.

11.8. Hướng dẫn:

- Khai báo một mảng chứa các số thực.
- Nhập phần tử vào cho mảng.
- Nhập giá trị cần so sánh với các giá trị của mảng.

- Duyệt mảng và lần lượt so sánh các giá trị của mảng với giá trị cho trước nếu có thì tăng biến đếm lên một đơn vị, ban đầu biến đếm có giá trị bằng 0.

11.9. Hướng dẫn:

- Khai báo mảng a để nhập dữ liệu. Mảng b làm trung gian.
- Nhập dãy số và lưu vào mảng a.
- Sử dụng vòng lặp For gán giá trị của các phần tử của mảng a cho các phần tử của mảng b:

For i:= k+1 to n do b[i-k]:= a[i];

For i:= 1 to k do b[i+k]:= a[i];

- Gán lại a:= b;

11.10. Hướng dẫn:

- Khai báo một mảng để chứa các phần tử của dãy FIBONACCI.
- Khởi gán: a[1]:=1; a[2]:= 1;
- Dùng lệnh for để tính a[i]:= a[i-1]+a[i-2] (i=3,4,5,...n).
- Thông báo kết quả a[n].

Phần nâng cao

11.11. Hướng dẫn:

- Nhập dãy số.
- Khai báo một biến t để chứa độ dài tạm thời của các dãy con các phần tử 0 liên tiếp; biến max để chứa độ dài cực đại và khởi gán chúng bằng 0.
- Dùng lệnh for để duyệt lần lượt các phần tử của dãy.

+ Nếu a[i] <> 0 thì

so sánh nếu max < t thì gán max:=t và t:=0;

+ Ngược lại: a[i]:= 0 thì t:= t+1;

- Cuối cùng so sánh nếu max < t thì gán lại max:=t;
- Thông báo kết quả max.

11.12. Hướng dẫn:

a.- Nhập mảng n phần tử.

- Nhập vị trí cần chèn k và giá trị cần chèn x.
- Giữ nguyên các phần tử từ 1 đến k-1.
- Dùng lệnh for gán các giá trị từ a[n+1] đến a[k+1]:

For i:= n+1 downto k+1 do a[i]:=a[i-1];

- Gán a[k]:= x;
- Thông báo kết quả.

b. - Nhập mảng n phần tử.

- Nhập vị trí cần xóa k.

- Giữ nguyên các phần tử từ 1 đến $k-1$.
- Dùng lệnh for gán các giá trị từ $a[k]$ đến $a[n-1]$:

For $i := k+1$ to n do $a[i-1] := a[i]$;

- Thông báo kết quả.

11.13. Hướng dẫn:

- Nhập dãy số.
- Khởi tạo biến đếm bằng 0 để đếm kết quả.
- Dùng vòng lặp for để duyệt tất cả các phần tử của dãy:

Với mỗi i ta lại so sánh $a[i]$ với tất cả các phần tử đứng sau nó nếu xảy ra $a[i]$ trùng với một phần tử nào đó thì bỏ qua, nếu không có phần tử nào trùng thì tăng biến đếm lên 1, đồng thời ghi phần tử thứ $a[i]$ đó ra màn hình.

- Thông báo kết quả là biến đếm.

11.14. Hướng dẫn:

- Khai báo mảng gồm 6 phần tử kiểu word.

Type mang = array[1..6] of word;

- Khai báo hằng m kiểu mảng:

Const m: mang = (1,2,5,10,20,50,100);

- Khai báo biến t để nhập số tiền, biến i kiểu số nguyên.
 - Khai báo biến k kiểu mảng để nhập số tờ của từng loại tiền có, biến đếm kiểu mảng để ghi số tờ của từng loại tiền.
 - Nhập số tiền t .
 - Nhập số tờ các loại tiền có vào mảng k và tính tổng số tiền có được.
 - Nếu tổng số tiền $< t$ thì thông báo không đủ tiền trả và dừng.
 - Ngược lại, dùng vòng lặp for duyệt lùi i từ 6 xuống 1 và gán $dem[i] := 0$;
- Lặp lại chừng nào $t \geq m[i]$ và $k[i] > 0$ các lệnh sau:

+ $t := t - m[i]$;

+ $dem[i] := dem[i] + 1$;

+ $k[i] := k[i] - 1$;

- Nếu $t = 0$ thì thông báo số tờ từng loại tiền thông qua mảng dem , ngược lại thông báo không có đủ tiền lẻ để trả.

11.15. Hướng dẫn:

- Nhập vào ma trận vuông $a[i,j]$ cấp n .
- Các phần tử nằm trên đường chéo chính là $a[i,i]$ ($i=1,2,\dots,n$)
- Các phần tử nằm phía trên đường chéo chính là $a[i,j]$ ($i=1..n, j>i$)
- Khai báo các biến $s1, s2$ để tính tổng.
- Thông báo kết quả.

11.16. Hướng dẫn:

- Khai báo 4 mảng hai chiều chứa các số nguyên.
- Nhập cấp của các ma trận vuông.
- Dùng hai vòng lặp for lồng nhau để nhập giá trị cho mảng
- Tính tổng của hai ma trận theo công thức.
- Tính tích của hai ma trận theo công thức.
- Thông báo kết quả ra màn hình.

11.17. Hướng dẫn:

- Nhập vào ma trận $a[i,j]$ cấp $m \times n$.
- Nhập k nguyên.
- Dùng hai vòng lặp for lồng nhau để duyệt tất cả các phần tử của ma trận và kiểm tra nếu thỏa mãn điều kiện $i-j=k$ thì ghi ra màn hình.

11.18. Hướng dẫn:

- Khởi tạo hàm randomize.
- Sử dụng vòng lặp for và khởi tạo hàm random(9) để lần lượt tạo các chữ số ngẫu nhiên cho từng giải và thông báo ra màn hình.

11.19. Hướng dẫn:

- Khai báo mảng.
- Nhập giá trị cho mảng và nhập giá trị x cho trước.
- Duyệt mảng và kiểm tra tổng giá trị của hai phần tử liên tiếp bằng x thì thông báo.

11.20. Hướng dẫn:

- Khai báo mảng hai chiều có m dòng và n cột.
- Duyệt hàng trước và với mỗi hàng tiến hành duyệt cột để tìm ra giá trị lớn nhất và nhỏ nhất (quy về mảng 1 chiều).
- Duyệt cột trước và với mỗi cột tiến hành duyệt hàng để tìm ra giá trị lớn nhất và nhỏ nhất (quy về mảng 1 chiều).
- Duyệt mảng và tìm giá trị lớn nhất, nhỏ nhất.
- Duyệt mảng và so sánh các giá trị với giá trị lớn nhất, nếu có thì tăng biến đếm, ban đầu biến đếm có giá trị bằng 0.

Bài 12. Kiểu xâu

I. Bài tập trắc nghiệm

12.1. B.	12.2. C.	12.3. C.	12.4. A.	12.5. B.
12.6. D.	12.7. D.	12.8. D.	12.9. C.	12.10. D.
12.11. C.	12.12. A.	12.13. B.	12.14. C.	12.15. A.
12.16. B.	12.17. C.	12.18. B.	12.19. A.	12.20. B.
12.21. C.	11.22. D.	11.23. B.	11.24. C.	11.25. B.

11.26. C.	11.27. C.	11.28. B.	11.29. C.	11.30. B.
11.31.A.	12.32. B.	12.33. C.	12.34. B.	12.35. C.
12.36. C.	12.37. D.	12.38. B.	12.39. B.	12.40. A.
12.41. D.	12.42. A.	12.43. C.	12.44. D.	12.45. D.
12.46. A.	12.47. A.	12.48. B.	12.49. B.	12.50. C.

II. Phần tự luận

12.1. Hướng dẫn:

- Khai báo biến st kiểu xâu để nhập dữ liệu, biến i điều khiển lệnh for.
- Nhập xâu st.
- Dùng lệnh for dạng lùi để ghi các kí tự của xâu từ sau tới.

12.2. Hướng dẫn:

- Khai báo biến st kiểu xâu để nhập dữ liệu, biến i điều khiển lệnh for.
- Nhập xâu st.
- Duyệt xâu với mỗi kí tự trên xâu thực hiện lệnh `st[i]:=upcase(st[i])`.
- Hiển thị xâu ra màn hình.

12.3. Hướng dẫn:

- Khai báo biến st kiểu xâu để nhập dữ liệu, biến i điều khiển lệnh for.
- Nhập xâu st.
- Duyệt xâu với mỗi kí tự trên xâu kiểm tra nếu thuộc khoảng ['A' .. 'Z'] thì thực hiện `st[i]:=chr(ord(st[i])+32)`.
- Hiển thị xâu ra màn hình.

12.4. Hướng dẫn:

- Khai báo biến st, st1 kiểu xâu để nhập dữ liệu và chứa xâu đảo, biến điều khiển vòng for i.
- Nhập xâu st.
- Khởi gán `st1:= ''`;
- Dùng lệnh for dạng lùi để lần lượt bổ sung các kí tự của xâu st từ sau tới cho xâu st1.
- So sánh hai xâu và thông báo kết quả.

12.5. Hướng dẫn:

- Khai báo
- Nhập xâu st.
- Dùng lệnh While kiểm tra vị trí đầu còn là kí tự trắng thì thực hiện xoá. (`While st[i]= #32 do delete(st,1,1)`)
- Dùng lệnh While kiểm tra vị trí cuối còn là kí tự trắng thì thực hiện xoá. (`While st[length(st)]= #32 do delete(st,length(st),1)`)

- Dùng lệnh While kiểm tra vị trí hai xâu trống liên tiếp nếu còn có thì thực hiện xoá. (While pos(#32#32,st)>0 do delete(st, pos(#32#32,st),1)).

12.6. Hướng dẫn:

- Khai báo xâu st.
- Nhập xâu st.
- Gán thêm kí tự trắng vào đầu xâu st.
- Khởi tạo biến đếm = 0 để đếm số từ.
- Dùng vòng lặp for để duyệt từ 1 đến length(st)-1 và kiểm tra nếu st[i] = #32 đồng thời st[i+1] <> #32 thì tăng biến đếm lên 1.
- Thông báo kết quả số từ là đếm.

12.7. Hướng dẫn:

- Khai báo biến.
- Nhập xâu họ và tên cho biến st.
- Xoá kí tự trắng dư thừa ở cuối xâu st.
- Gán i = length(st);
- Dùng vòng lặp While duyệt từ sau tới chừng nào mà st[i] còn khác kí tự trắng thì giảm i đi 1 cho đến khi gặp kí tự st[j] khác kí tự trắng đầu tiên thì dừng lại và thông báo tên gồm các kí tự từ vị trí j+1 đến hết xâu.

Phần nâng cao

12.8. Hướng dẫn:

- Khai báo
- Nhập số nguyên n.
- Khởi gán biến s để chứa dãy nhị phân.
- Lặp lại công việc sau chừng nào n>0:
 - + Nếu n chẵn thì s:= '0'+s; ngược lại n lẻ thì S:= '1'+s;
 - + Gán n:= n div 2;
- Thông báo kết quả s.

12.9. Hướng dẫn:

- Khai báo dữ liệu.
- Nhập xâu st bất kì.
- Khởi tạo giá trị cho biến i, biến dem để đếm số từ.
- Lặp lại công việc sau chừng nào i <= length(st):
 - + Lặp lại lệnh i:= i+1 chừng nào s[i] = ''
 - + Nếu i < length(st) thì:
 - . Chừng nào st[i] <> '' thì viết st[i] và tăng i lên 1.
 - . Tăng biến đếm lên 1.
 - . Xuống dòng.

+ Tăng i lên 1.

- Thông báo kết quả số từ là đếm.

12.10. Hướng dẫn:

- Khai báo các biến so , $so1$, $so2$, nho , i , k kiểu nguyên để chứa các số và biến điều khiển lệnh lặp.

- Khai báo các biến $s1$, $s2$, $ch1$, $ch2$, t , s kiểu xâu.

- Nhập số nguyên thứ nhất lưu vào xâu $s1$, số nguyên thứ 2 lưu vào xâu $s2$.

- Khởi gán $s := 0$;

- Nếu độ dài hai xâu $s1$ và $s2$ khác nhau thì bổ sung các kí tự '0' vào trước xâu ngắn để được hai xâu có độ dài bằng nhau.

- Dùng vòng lặp for duyệt từ cuối chuỗi đến đầu chuỗi và thực hiện phép cộng với các cặp kí tự số tương ứng bằng cách:

+ Dùng thủ tục val đổi kí tự $s1[i]$ thành số $so1$, kí tự $s2[i]$ thành số $so2$.

+ Gán $so := so1 + so2 + nho$

+ $so := so \bmod 10$

+ $nho := so \text{ div } 10$

+ Dùng thủ tục str đổi số thành xâu và gán vào biến t .

+ Gán $s := t + s$;

- Nếu $nho > 0$ thì gán $s := '1' + s$;

- Thông báo kết quả là s .

12.11. Hướng dẫn:

- Khai báo.

- Nhập danh sách họ tên.

- Duyệt danh sách và sử dụng giải thuật so sánh để hoán đổi các giá trị cho nhau.

- In danh sách ra màn hình.

12.12. Hướng dẫn:

- Khai báo.

- Nhập vào hai từ

- Sắp xếp các kí tự trong cả hai chuỗi theo cùng một thứ tự tăng dần hay giảm dần, sau đó so sánh hai chuỗi đó với nhau, nếu bằng nhau thì chúng là anagram của nhau, nếu khác thì chúng không phải là anagram của nhau. Ví dụ: dear, read, dare là các anagram của nhau.

Bài 13. Kiểu Bản ghi (record)

I. Bài tập trắc nghiệm

- 13.1. D. 13.2. B. 13.3. C. 13.4. A. 13.5. C.
13.6. A. 13.7. B. 13.8. B. 13.9. C. 13.10. D.
13.11. B. 13.12. D. 13.13. A. 13.14. D. 13.15. B. 13.16. A.

II. Bài tập tự luận

13.1. Hướng dẫn:

- Sử dụng cú pháp định nghĩa kiểu để khai báo.

13.2. Hướng dẫn:

- Khai báo kiểu phân số như sau:
Type Phanso = Record
 Tuso, mauso: Word;
 End;
- Khai báo các biến khác
- Sử dụng các giải thuật cộng, trừ, nhân, chia phân số để thực hiện tính toán.

13.3. Hướng dẫn:

- Khai báo kiểu bản ghi khách hàng gồm các trường đã cho.
- Khai báo các biến khác.
- Nhập dữ liệu.
- Tính chỉ số công tơ tiêu thụ của mỗi khách hàng
- Kiểm tra số tiêu thụ, tùy theo giá trị cụ thể để tính theo các mức giá khác nhau.

13.4. Hướng dẫn:

- Khai báo kiểu bản ghi học sinh gồm các trường đã cho.
- Khai báo một mảng mà mỗi phần tử là một bản ghi học sinh.
- Khai báo các cột điểm có kiểu dữ liệu phù hợp.
- Nhập danh sách học sinh và tiến hành tính điểm trung bình kiểm tra và điểm trung bình.
- Duyệt danh sách và kiểm tra học sinh nào có điểm trung bình dưới 5 thì thông báo thi lại.

13.5. Hướng dẫn:

- Khai báo bản ghi học sinh gồm hai trường.
- Khai báo mảng chứa danh sách học sinh.
- Nhập danh sách học sinh.
- Sắp xếp danh sách học sinh.

- Xếp hạng cho học sinh.
- Hiện thị kết quả ra màn hình.

13.6. Hướng dẫn

- Khai báo dữ liệu kiểu bản ghi, khai báo mảng chứa danh sách thí sinh, mỗi thí sinh là một bản ghi.
- Nhập dữ liệu vào
- Sử dụng giải thuật sắp xếp để sắp xếp danh sách.
- Hiện thị danh sách ra màn hình.
- Hiện thị danh sách trúng tuyển ra màn hình. Sử dụng hàm and để lấy điều kiện.

13.7. Hướng dẫn:

- Khai báo biến kiểu bản ghi
- Khai báo mảng chứa danh sách các bản ghi
- Nhập danh sách giáo viên, chú ý kiểm tra họ tên khác rỗng thì mới nhập các thông tin khác.
- Sắp xếp danh sách theo trật tự giảm dần.
- Kiểm tra các giáo viên thoả mãn điều kiện nghỉ hưu rồi thông báo ra màn hình. Chú ý sử dụng từ khoá OR để kiểm tra hoặc giáo viên nam, hoặc giáo viên nữ.

13.8. Hướng dẫn:

- Khai báo kiểu bản ghi xe máy.
- Khai báo mảng chứa các xe máy, mỗi phần tử là một bản ghi.
- Nhập danh sách các xe máy, chú ý không nhập thuế mà được tính bằng công thức.
- Duyệt danh sách và kiểm tra xe máy nào có nguồn gốc từ Nhật Bản thì hiện thị ra.
- Tìm xe máy có giá cao nhất, nếu tìm thấy thì hiện thị các thông tin biết tên xe, kí hiệu xe và màu.

13.9. Hướng dẫn:

- Khai báo kiểu đường thẳng gồm các thông tin toạ độ x_1, y_1 , hệ số góc.
- Nhập toạ độ và hệ số góc cho đường thẳng thứ nhất, thứ hai.
- Kiểm tra điều kiện song song theo điều kiện.
- Kiểm tra điều kiện cắt nhau hay không và kiểm tra chúng có vuông góc hay không theo điều kiện.

13.10.

Hướng dẫn:

- Xem hướng dẫn bài 13.2

- Kết hợp theo các điều kiện cụ thể.

Chương 5. TỆP VÀ THAO TÁC VỚI TỆP

Bài 14. Kiểu dữ liệu tệp.

I. Bài tập trắc nghiệm

14.1. A. 14.4. C. 14.2. B. 14.5. D. 14.3. A.

II. Bài tập tự luận: Xem lí thuyết để trả lời các câu hỏi.

Bài 15. Thao tác với tệp

I. Bài tập trắc nghiệm

15.1. A. 15.2. D. 15.3. B. 15.4. A. 15.5. D.
 15.6. C. 15.7. C. 15.8. D. 15.9. C. 15.10. C.
 15.11. C 15.12. A 15.13. D. 15.14. B. 15.15. A
 15.16. C. 15.17. C. 15.18. D. 15.19. C. 15.20. C.

II. Bài tập tự luận

15.1. Hướng dẫn:

- Sử dụng từ khoá Type để khai báo một kiểu dữ liệu mới.

15.2. Hướng dẫn:

- Sử dụng từ khoá Var để khai báo biến dạng file văn bản.

15.3. Hướng dẫn:

- Sử dụng từ khoá Type để khai báo một kiểu bản ghi.
- Khai báo một kiểu file mà kiểu phần tử của file là kiểu bản ghi.

15.4. Hướng dẫn:

- Khai báo dữ liệu
- Đưa dữ liệu vào cho file là các số nguyên từ 1 đến 100, tên file là songuyen.dat.

15.5. Hướng dẫn:

- Khai báo dữ liệu
- Đưa dữ liệu vào cho file là các số nguyên, file có tên là songuyen.dat.
- Sử dụng giải thuật sắp xếp để sắp xếp các số trong file.

15.6. Hướng dẫn:

- Khai báo dữ liệu
- Tạo hai file và sao chép dữ liệu từ file này qua file khác.

15.7. Hướng dẫn:

- Sử dụng hàm seek(tên file, k) để định vị con trỏ file đến vị trí cần kiểm tra, sau đó xử lí.

Bài 16. Ví dụ làm việc với tệp.

I. Bài tập trắc nghiệm

16.1. B 16.3. C. 16.7. C. 16.9. C. 16.5. C. 16.11. A.
16.2. C. 16.4. A. 16.8. A. 16.10. C. 16.6. B.

II. Bài tập tự luận

16.1. Hướng dẫn

- Khai báo dữ liệu:
- Sử dụng hàm ngẫu nhiên Random để lấy số.
- Lặp cho đến khi số nhận được số bằng 0 thì thoát.
- Sử dụng giải thuật sắp xếp để sắp xếp dữ liệu, có thể sắp xếp trực tiếp hoặc dùng mảng phụ.
- Hiện thị kết quả ra màn hình.

16.2. Hướng dẫn

- Khai báo dữ liệu
- Tạo file mới, nhập dữ liệu vào, việc nhập kết thúc khi họ tên là xâu rỗng (sử dụng vòng lặp không xác định).
- Sử dụng giải thuật sắp xếp để sắp xếp danh sách.
- Hiện thị danh sách ra màn hình.
- Hiện thị danh sách trùng tuyến ra màn hình. Sử dụng hàm and để lấy điều kiện.

16.3. Hướng dẫn

- Khai báo tập tin văn bản.
- Kiểm tra chưa kết thúc tập tin thì đọc từng dòng của tập tin vào một biến có kiểu string, với mỗi dòng đọc được sử dụng một biến đếm và tăng giá trị lên 1 đồng thời đếm số từ trong đó rồi cộng dồn số từ.

16.4. Hướng dẫn

- Khai báo dữ liệu.
- Xử lý tập tin như sau: Nhập vào một số và ghi vào tập tin, khi ghi vào thì ghi thêm vào sau đó một kí tự trắng. Nếu số lượng các số được ghi vào file là chia hết cho 10 thì xuống dòng.

16.5. Hướng dẫn:

- Khai báo dữ liệu.
- Khi xử lý tập tin chú ý mỗi dòng chỉ có đúng 10 số, ngoại trừ dòng cuối cùng.

16.6. Hướng dẫn:

- Khai báo dữ liệu.

- Đọc từng dòng trên tập tin đã có vào biến có kiểu xâu .
- Xử lý lại xâu theo yêu cầu.
- Ghi xâu đã xử lý vào tập tin tạm thời đã mở mới để ghi.
- Sau khi thực hiện xong công việc trên thì xóa tập tin đã có và đổi tên tập tin tạm thời thành tập tin đã có.

16.7. Hướng dẫn:

- Khai báo dữ liệu.
- Xử lý tập tin
- Sử dụng một biến để quản lý số dòng, dòng đầu tiên gán bằng 1, kiểm tra chưa kết thúc file thì đọc từng dòng, sau mỗi lần đọc thì gán số dòng và tăng giá trị lên một đơn vị.

16.8. Hướng dẫn:

- Khai báo dữ liệu.
- Xử lý tập tin
- Đọc file, kiểm tra các kí tự đọc được có thuộc khoảng 'A'..'Z' và 'a'..'z' hay không, nếu có thì chuyển sang kí tự hoa và tăng tần suất xuất hiện lên.

16.9. Hướng dẫn:

- Khai báo dữ liệu.
- Xử lý tập tin
- Mở hai tập tin sau đó đọc từ tập tin này sang tập tin khác.

16.10. Hướng dẫn:

- Khai báo dữ liệu kiểu bản ghi.
- Khai báo file có kiểu phần tử là các bản ghi.
- Sử dụng vòng lặp không xác định nhập họ tên với điều kiện thoát là xâu rỗng.
- Mở file và thực hiện bổ sung thêm phần tử.
- Sử dụng giải thuật sắp xếp để sắp xếp tên theo alphabet.
- Tìm cách học sinh thi lại và hiển thị ra màn hình.

Chương 6. CHƯƠNG TRÌNH CON VÀ LẬP TRÌNH CÓ CẤU TRÚC

Bài 17. Chương trình con và phân loại.

Bài tập trắc nghiệm

- | | | | | |
|----------|----------|----------|----------|-----------|
| 17.1. D. | 17.2. B. | 17.3. C. | 17.4. D. | 17.5. D. |
| 17.6. D. | 17.7. B. | 17.8. C. | 17.9. C. | 17.10. C. |

Bài 18. Ví dụ về cách viết và sử dụng chương trình con.

I. Bài tập trắc nghiệm

- | | | | | |
|-----------|-----------|-----------|-----------|-----------|
| 18.1. D. | 18.2. B. | 18.3. A. | 18.4. C. | 18.5. B. |
| 18.6. D. | 18.7. B. | 18.8. B. | 18.9. C. | 18.10. B. |
| 18.11. D. | 18.12. D. | 18.13. C. | 18.14. A. | 18.15. B. |
| 18.16. C. | 18.17. B. | 18.18. C. | | |

II. Bài tập tự luận

18.1. Hướng dẫn:

- Khai báo đầu hàm như sau: Function log(a, b:word): real;

18.2. Hướng dẫn:

- Khai báo đầu hàm như sau: Function max(a, b: Real): real;

18.3. Hướng dẫn:

- Khai báo đầu hàm như sau: Function BCNN(a, b: word): word;

18.4. Hướng dẫn:

Có hai cách:

- Có thể dùng thủ tục STR để chuyển số sang dạng chuỗi, tiếp theo đảo ngược chuỗi tìm được và sau cùng chuyển đổi chuỗi đảo ngược thành dạng số bởi thủ tục val. Đó chính là giá trị trả về của hàm.
- Nếu không dùng kiểu chuỗi thì lần lượt lấy các chữ số của n theo thứ tự từ trái sang phải để lắp ghép thành các chữ số đảo ngược.

18.5. Hướng dẫn:

- Để kiểm tra một số n có phải là số nguyên tố hay không thì kiểm tra n có chia hết cho một trong các số từ 2 đến phần nguyên của căn bậc hai của n hay không? Nếu đúng thì n không phải là số nguyên tố, nếu sai thì n là số nguyên tố.

18.6. Hướng dẫn:

- Sử dụng giải thuật bài 18.6, chú ý phần trình bày mỗi dòng có 10 số.

18.7. Hướng dẫn:

- Không đệ quy: Dùng vòng lặp For

- Đệ quy: sử dụng công thức sau:
$$S(n) = \begin{cases} 1 & \text{nếu } n = 1 \\ S(n-1) + \text{sqr}(n) & \text{nếu } n > 1 \end{cases}$$

18.8. Hướng dẫn:

- Không dùng đệ quy: Dùng vòng lặp For.

- Dùng đệ quy: sử dụng công thức

$$S = x \sin x (1 + x \sin x + x^2 \sin^2 x + \dots + x^{n-1} \sin^{n-1} x)$$

Phần nâng cao

18.9. Hướng dẫn:

- Nhập chuỗi

- Dùng lệnh lặp While `st[1]!='\0' do delete(st,1,1)` để xoá các kí tự trắng đầu xâu.
- Dùng lệnh lặp While `st[length(st)]!='\0' do delete(st,length(st),1)` để xoá các kí tự trắng cuối xâu.
- Dùng lệnh lặp While `pos('\0\0',st)>0 do delete(st,pos('\0\0',st),1)` để xoá các kí tự trắng đi liền nhau ở giữa xâu.

18.10. Hướng dẫn:

- Sử dụng hàm Uppcase kết hợp với vòng For

18.11. Hướng dẫn:

- Dành cho bạn đọc

18.12. Hướng dẫn:

- Sử dụng vòng lặp for để đếm

18.13. Hướng dẫn:

- Sử dụng một số thủ tục chuẩn của chuỗi

18.14. Hướng dẫn:

- Dành cho độc giả.

18.15. Hướng dẫn:

- Sử dụng hàm pos để xác định vị trí xâu s1 xuất hiện trong xâu s
- Nếu vị trí bằng 0 thì kết thúc vì không tìm thấy
- Nếu vị trí khác không thì xoá xâu s1 trong xâu s, tại vị trí đã tìm thấy, sau đó tại vị trí này chèn xâu s2 vào. Lặp lại quá trình trên.

18.16. Hướng dẫn:

- Dùng thủ tục readln để nhập xâu lưu vào biến s. Nếu không hợp lệ thì yêu cầu nhập lại cho đến chừng nào đúng.

18.17. Hướng dẫn:

- Ban đầu cho xâu s bằng rỗng, người sử dụng nhập từng kí tự của xâu bằng hàm readkey, nếu kí tự nhập vào hợp lệ thì cộng vào phía phải của xâu s và cho hiện kí tự đó ra màn hình, rồi nhập lại. Đến chừng nào kí tự nhập vào là #13 thì kết thúc.

18.18. Hướng dẫn:

- Khai báo kiểu bản ghi học sinh gồm các trường đã cho.
- Khai báo một mảng mà mỗi phần tử là một bản ghi học sinh.
- Khai báo các cột điểm có kiểu dữ liệu phù hợp.
- Nhập danh sách học sinh và tiến hành tính điểm trung bình kiểm tra và điểm trung bình.
- Duyệt danh sách và kiểm tra học sinh nào có điểm trung bình dưới 5 thì thông báo thi lại.

Chương 7

MỘT SỐ BÀI TOÁN NÂNG CAO, CÁC BÀI THỰC HÀNH DÙNG TRÊN MÁY VÀ MỘT SỐ ĐỀ KIỂM TRA DÙNG THAM KHẢO

A. Kiến thức mở rộng

ĐỒ HOẠ VÀ ÂM THANH

§1. MỘT SỐ YẾU TỐ ĐỒ HOẠ

1. Các thiết bị và chương trình hỗ trợ đồ hoạ

Màn hình có thể làm việc trong hai chế độ: Chế độ văn bản và chế độ đồ hoạ. Có thể hình dung màn hình như là một bảng các điểm sáng. Hình ảnh đồ hoạ được xây dựng từ các điểm sáng được gọi là các điểm ảnh (pixel). Pixel là đơn vị cơ sở của màn hình đồ hoạ.

Card màn hình là thiết bị đảm bảo tương tác giữa bộ xử lí và màn hình để thực hiện các chế độ phân giải và màu sắc. Tên gọi của card màn hình thường trùng với tên của loại màn hình, ví dụ VGA, SVGA,...

TP cung cấp các chương trình điều khiển tương ứng với các loại card đồ hoạ. Các chương trình điều khiển này nằm trong các tệp có phần mở rộng là BGI (Borland Graphics Interface). Muốn hoạt động trong chế độ đồ hoạ, cần phải có tệp BGI thích hợp với màn hình đang dùng. Các tệp BGI được TP ngầm định để trong thư mục con BGI của thư mục cài đặt nó. Khi khởi động đồ hoạ, cần chỉ rõ đường dẫn đến các tệp này.

Toạ độ màn hình đồ hoạ được đánh số từ 0, cột được tính từ trái sang phải và dòng được tính từ trên xuống dưới. Giá trị lớn nhất của toạ độ dòng và cột được gọi là độ phân giải của màn hình. Thông thường TP xác lập độ phân giải là 640x480.

Để thực hiện các chức năng đồ hoạ cần sử dụng các thủ tục và hàm trong thư viện có tên là GRAPH.

2. Khởi tạo chế độ đồ hoạ

Một chương trình đồ hoạ bao giờ cũng mở đầu bằng việc khởi động chế độ đồ hoạ. Các thủ tục, hàm của thư viện GRAPH luôn ngầm định là chỉ hoạt động khi chế độ đồ hoạ đã được thiết lập. Cần phải quay về chế độ màn hình văn bản, khi kết thúc chương trình đồ hoạ.

Thủ tục cần sử dụng để khởi động chế độ đồ hoạ là:

Procedure InitGraph(var driver, mode: integer; path: string);

trong đó :

- Driver là số hiệu của trình điều khiển BGI.
- Mode là số hiệu của độ phân giải.
- Path là đường dẫn đến các tệp BGI.

Thông thường, ta nên sử dụng cách khởi động tự động chế độ đồ hoạ sau đây: biến driver được gán giá trị 0 (mà GRAPH định sẵn một tên là Detect)

Ví dụ:

Giả sử màn hình làm việc là VGA và các tệp BGI đang để ở thư mục C:\TP\BGI. Khi đó :

Driver := 0;

InitGraph(driver, mode, 'C:\TP\BGI'); sẽ khởi động đồ hoạ với chế độ VGAHi.

Sau khi kết thúc các chương trình đồ hoạ để trở về chế độ màn hình văn bản cần gọi thực hiện thủ tục:

Procedure CloseGraph;

3. Các thủ tục vẽ điểm và đường

Vẽ điểm và đường là hai thao tác cơ bản của đồ hoạ. Trong TP vẽ điểm được thực hiện nhờ gọi thủ tục:

PutPixel(X, Y : integer; color : word);

trong đó :

- X và Y là toạ độ của điểm.
- Color là màu của nó.

Các đoạn thẳng được xác định bởi toạ độ của hai đầu mút. Để vẽ đoạn thẳng nối hai điểm sử dụng thủ tục: Line(x1, y1, x2, y2 : integer);

Trong đó (x1, y1) và (x2, y2) là các toạ độ của hai đầu mút của đoạn thẳng.

Khi cần phải đặt màu cho nét vẽ, trước khi vẽ cần gọi thủ tục :

Procedure SetColor(color: word);

Để vẽ các đoạn thẳng còn có thể sử dụng hai thủ tục sau đây :

- Procedure LineTo(x, y: integer); thực hiện nối điểm tại vị trí con trỏ màn hình đồ hoạ đến điểm có toạ độ (x,y).
- Procedure LineRel(dx, dy: integer); thực hiện nối điểm tại vị trí con trỏ màn hình đồ hoạ đến điểm có toạ độ bằng toạ độ của vị trí hiện tại cộng với gia số (dx,dy).

Ví dụ:

Chương trình sau đây vẽ các đường gấp khúc “ngẫu nhiên” nhờ thủ tục LineTo, mỗi đoạn có một màu ngẫu nhiên. Vị trí bắt đầu vẽ là tâm của màn hình. Kết thúc việc vẽ bằng cách nhấn một phím bất kì.

Uses crt,graph;

Function DetectInit(path : string): integer;

Var drive, mode : integer;

Begin

drive := 0;

InitGraph(drive, mode, path);

DetectInit := GraphResult;

End;

Begin

If DetectInit('C:\TP\BGI') <> 0 then

Begin

Write('Loi do hoa! An phim Enter de ket thuc...');

Readln;

End

Else

Begin

Randomize;

MoveTo(Getmaxx div 2, Getmaxy div 2);

Repeat

SetColor(Random(GetMaxColor));

{ Thiết lập màu một cách ngẫu nhiên }

LineTo(Random(Getmaxx), Random(Getmaxy));

Delay(200); { Tạm dừng }

Until Keypressed;

End

End;

Trong chương trình này sử dụng các hàm:

+ Function GetMaxX: integer;

+ Function GetMaxY: integer;

để xác định giá trị lớn nhất có thể của tọa độ X và Y của màn hình (kích thước màn hình trong chế độ đồ họa đang sử dụng).

Thủ tục: Procedure MoveTo(x, y: integer); thực hiện di chuyển con trỏ đến vị trí tọa độ (x,y).

4. Một số thủ tục vẽ hình đơn giản

Dưới đây giới thiệu một số thủ tục thực hiện vẽ các đường khép kín:

- Procedure Circle(x, y : integer; r : word);
vẽ vòng tròn tâm tại (x,y) bán kính r;
- Procedure Ellipse(x,y : integer; StAngle,EndAngle,Xr,Yr: word); vẽ
cung của ellip tâm tại điểm (x,y), với các bán kính trục Xr, Yr từ góc
khởi đầu StAngle đến góc kết thúc EndAngle.
- Procedure Rectangle(x1,y1,x2,y2 : integer);
vẽ hình chữ nhật có các cạnh song song với các trục toạ độ, trong đó
(x1,y1) là toạ độ của đỉnh trái trên còn (x2,y2) là toạ độ của đỉnh phải dưới.

Ví dụ:

Chương trình dưới đây minh hoạ cho việc sử dụng các hàm trên.

Program GraphDemo;

Uses

crt, graph;

Var

gd, gm : integer;

xm, ym, xmaxD4, ymaxD4 : word;

BEGIN

gd:= detect;

Initgraph(gd, gm, 'C:\TP\BGI');

xm:=GetmaxX; ym:= GetmaxY;

{ Vẽ hình chữ nhật với nét vẽ màu vàng đóng khung màn hình }

SetColor(Yellow);

Rectangle(0,0,xm,ym); readln;

{ Vẽ vòng tròn màu xanh lá cây tâm tại(450, 100) bán kính 50 }

SetColor(LightGreen);

Circle(450, 100, 50); readln;

SetColor(Red); { Vẽ ellip màu đỏ }

Ellipse(100, 200, 0, 360, 50,120);

Readln;

CloseGraph;

END.

§2. ÂM THANH

Khi được bật loa của máy tính sẽ phát ra âm thanh ở một tần số nào đó cho đến khi được tắt. Để khai thác khả năng này, thư viện CRT cung cấp thủ tục : Procedure Sound(h : word); làm cho loa phát ra một âm thanh có tần số là h (tính theo đơn vị Hertz) và thủ tục: Procedure NoSound; tắt âm thanh đang được phát bởi loa.

Mỗi khi gọi thủ tục Sound để phát một âm thanh nào đó, âm thanh này sẽ được kéo dài cho đến khi gặp lời gọi NoSound. Để xác định khoảng thời gian kéo dài, ta dùng thủ tục Delay với tham trị t. Thủ tục sau phát một âm thanh với cao độ h và trường độ t:

```
Procedure CreatSound(h, t : word);  
  Begin  
    Sound(h); Delay(t); NoSound;  
  End;
```

1. Mô phỏng âm thanh

Có thể mô phỏng một âm thanh nhờ thủ tục CreatSound(h, t). Vấn đề còn lại là tìm các giá trị thích hợp của h và t.

Ví dụ:

Chương trình dưới đây tạo tiếng lộc cộc của vó ngựa phi:

```
Program DemoSound;  
Uses CRT;  
Procedure CreatSound(h, t: word);  
  Begin  
    Sound(h); Delay(t); NoSound;  
  End;  
Procedure Clockoc(h: word);  
  Begin  
    CreatSound(h, 20);  
    Delay(50);  
    CreatSound(h, 20);  
    Delay(100);  
    CreatSound(h, 40);  
  End;  
Begin  
  Repeat  
    Clockoc(500);
```



```
Delay(200);  
Until KeyPressed;  
End.
```

Việc khai thác âm thanh thường được dùng để xây dựng các chương trình trò chơi hoặc mô phỏng. Vì thế nếu thường xuyên phải phát triển những ứng dụng loại này trên TP thì nên tổ chức riêng một thư viện, chứa một số âm thanh mẫu như tiếng các loại động cơ, tiếng còi, tiếng nổ, tiếng va chạm, ... để tiện dùng.

2. Mô phỏng nốt nhạc

Một trường hợp riêng của việc khai thác âm thanh là mô phỏng các nốt nhạc. Một nốt nhạc cao độ h , trường độ t sẽ được thực hiện bởi lời gọi :

```
CreatSound(h, t).
```

Theo âm luật, từ một quãng 8 đến quãng 8 kế tiếp, thừa số nhân của cao độ là 2. Trong một quãng 8, có 12 khoảng nửa cung, vì thế thừa số nhân giữa hai khoảng này là căn bậc 12 của 2. Vì vậy chỉ cần biết cao độ của một nốt làm gốc, ta có thể tính cao độ của các nốt khác. Chẳng hạn, xuất phát từ cao độ của nốt đô trung là 512, cao độ của các nốt trong quãng 8 trung lần lượt là (sau khi đã làm tròn): Đô trung = 512, đô thăng = 542, rê = 575, rê thăng = 609, mi = 645, fa = 683, fa thăng = 724, son = 767, son thăng = 813, la = 861, la thăng = 912, si = 967. Về trường độ, lấy nốt móc đơn làm đơn vị là 150, ta tính được các trường độ khác: Nốt đen = 300, nốt trắng = 600, nốt trắng chấm = 900, nốt tròn = 1200, ...

Ví dụ:

Chương trình dưới đây thể hiện một bản nhạc ngắn của đồng hồ, tất cả các nốt đều ở quãng 8 trung:

```
Program DemoMusic;
```

```
Uses CRT;
```

```
Procedure CreatSound(h, t: word);
```

```
Begin
```

```
Sound(h); Delay(t); NoSound;
```

```
End
```

```
Begin
```

```
CreatSound(967, 300); {si đen}
```

```
CreatSound(767, 300); {son đen}
```

```
CreatSound(861, 300); {la đen}
```

```
CreatSound(575, 900); {rê trắng chấm}
```

```
Delay 60);
```

```

CreatSound(575, 300); {rê đen}
CreatSound(861, 300); {la đen}
CreatSound(967, 300); {si đen}
CreatSound(767, 900); {son trắng chấm}
Delay(60);
CreatSound(967, 300); {si đen}
CreatSound(861, 300); {la đen}
CreatSound(767, 300); {son đen}
CreatSound(575, 900); {rê trắng chấm}
Delay(60);
CreatSound(575, 300); {rê đen}
CreatSound(861, 300); {la đen}
CreatSound(967, 300); {si đen}
CreatSound(767, 900); {son trắng chấm}
End.

```

B. Các bài thực hành mẫu

- Nhập chương trình
- Chạy và kiểm thử
- Đề xuất phương án sửa lỗi, hoàn thiện chương trình

§1. BÀI THỰC HÀNH SỐ 1

1) Soạn thảo chương trình giải phương trình bậc 2, chạy và nhận xét:

$$AX^2 + BX + C = 0 \quad \text{với } A \neq 0.$$

```

PROGRAM G_P_T_B2;
USES CRT;
VAR A, B, C, D : REAL;
    X1, X2 : REAL;
BEGIN
  CLRSCR;
  WRITE(' NHAP GIA TRI A, B, C: ');
  READLN(A,B,C);
  D := SQRT(B*B - 4*A*C);
  X1 := (-B - D)/(2*A);
  X2 := -B/A - X1;

```

```
WRITE('X1 = ', X1:6:2,' X2 = ',X2:6:2);
```

```
READLN
```

```
END.
```

2) Gõ *F2* và lưu chương trình với tên là PHTR2.PAS.

3) Gõ *Alt + F9* để dịch và sửa lỗi cú pháp (nếu có).

4) Gõ *Ctrl + F9* để thực hiện chương trình, nhập vào các giá trị 1 -3 và 2.
Quan sát kết quả trên màn hình, nhận xét.

5) Gõ *Ctrl + F9* và nhập các giá trị 1 0 -2.

Quan sát kết quả trên màn hình, nhận xét.

6) Sửa lại chương trình trên sao cho không cần dùng biến trung gian D. Sau khi sửa, thực hiện lại chương trình với các bộ dữ liệu trên.

7) Sửa lại chương trình đã cho bằng cách thay đổi công thức tính X2 (còn có 2 cách tính X2 theo công thức khác).

8) Thực hiện chương trình với bộ dữ liệu 1 -5 6 (Xem kết quả nhận được và nhận xét).

9) Thực hiện chương trình với bộ dữ liệu 1 1 1, quan sát cách thông báo lỗi của TP. Sửa chương trình lại cho đúng, lưu lại tên mới và chạy kiểm tra.

§2. BÀI THỰC HÀNH SỐ 2

Bài toán

Lập trình giải và biện luận hệ phương trình bậc nhất hai ẩn:

$$\begin{cases} ax + by = e \\ cx + dy = f \end{cases}$$

Thuật giải : Giải bằng phương pháp tính định thức.

a) Gõ chương trình :

```
PROGRAM HE_PT;
```

```
USES CRT;
```

```
VAR A,B,C,D,E,F : REAL;
```

```
DT,DTX,DTY,X,Y : REAL;
```

```
BEGIN
```

```
CLRSCR;
```

```
WRITE(' A,B,C : '); READLN(A,B,C);
```

```
WRITE(' D,E,F : '); READLN(D,E,F);
```

```
DT := A*E - D*B;
```

```

DTX := C*E - F*B;
DTY := A*F - D*C;
IF DT = 0 THEN
    WRITELN(' PT Vo dinh hoac vo nghiem ')
ELSE
    BEGIN
        WRITELN(' He phuong trinh co hai nghiem : ');
        WRITELN(' X = ', DTX/DT:10:5);
        WRITELN(' va Y = ', DTY/DT:10:5)
    END;
READLN;
END.

```

b) Lưu chương trình lên đĩa với tên HEPT;

c) Bấm phím *F7* để thực hiện từng câu lệnh chương trình, nhập các giá trị
 $A = 4, B = -3, C = -7, D = -2, E = 7, F = 31$;

d) Vào mục chọn Debug mở cửa sổ hiệu chỉnh để xem giá trị DT, DTX, DTY.

e) Gõ phím *F7* để thực hiện các câu lệnh tính những giá trị nói trên, so sánh với kết quả chờ đợi phải có ($DT = 22, DTX = 44, DTY = 110$).

f) Quan sát quá trình rẽ nhánh.

g) Lặp lại các bước trên với bộ dữ liệu $A = 5, B = 6, C = 7, D = 15, E = 18, F = 45$;

h) Cải tiến chương trình: Sửa đổi cách kiểm tra điều kiện và tổ chức rẽ nhánh, hiệu chỉnh chương trình nhận được. Chạy và kiểm tra lại chương trình.

§3. BÀI THỰC HÀNH SỐ 3

Bài 1. Cho mảng A gồm n ($n \leq 100$) số nguyên, mỗi số theo trị tuyệt đối không vượt quá 300. Tính tổng các phần tử của mảng là bội số của một số nguyên dương k cho trước.

a) Hãy tìm hiểu, gõ và chạy thử chương trình sau đây :

```

Program      Sum1;
Uses crt;
Const nmax=100;
Type MyArray=Array[1..nmax] Of Integer;
Var  a : MyArray;

```

```

    s,n,i,k: Integer;
Begin
    Clrscr; Randomize;
    Write('Nhap n ='); ReadLn(n);
    {Tạo ngẫu nhiên mảng gồm n số nguyên}
    For i:= 1 to n do A[i]:= Random(301)- Random(301);
    Write('Nhap k = ');
    ReadLn(k); s:=0;
    For i:=1 To n Do
        If A[i] mod k = 0 then s:=s+A[i];
    WriteLn('Tong can tinh la: ',s);
    ReadLn;
End.

```

b) Hãy đưa các câu lệnh sau đây vào những vị trí cần thiết nhằm sửa đổi chương trình trong câu a để có được chương trình đưa ra số lượng số dương và số lượng số âm trong mảng.

```

Posi, neg : Integer;
posi:= 0; neg:= 0;
If A[i]>0 Then Inc(posi)
    Else If A[i]<0 Then Inc(neg);
WriteLn(posi:4,neg:4);

```

Bài 2. Viết chương trình tìm phần tử có giá trị lớn nhất của mảng. Đưa ra màn hình chỉ số và giá trị của phần tử tìm được. Nếu có nhiều phần tử có cùng giá trị lớn nhất thì đưa ra phần tử với chỉ số nhỏ nhất.

a) Hãy gõ, tìm hiểu chương trình sau đây :

```

Program MaxElement;
Const Nmax= 100;
Type MyArray = Array[1..Nmax] Of Integer;
Var a : MyArray;
    n,i,j : Integer;
Begin
    Randomize;
    Write('Nhap n = '); ReadLn(n);

```



```

    {Tạo ngẫu nhiên mảng gồm n số nguyên}
    For i:= 1 to n do A[i]:= Random(300)- Random(300);
    j:= 1;
    For i:= 2 To n Do If A[i] > A[j] Then j:= i;
    Write ('Chi so: ',j, ' Gia tri: ',A[j]:4);
    ReadLn;
End.

```

b) Viết lại chương trình trên để đáp ứng yêu cầu sau: Đưa ra các chỉ số của các phần tử có cùng giá trị lớn nhất.

§4. BÀI THỰC HÀNH SỐ 4

Bài 1.

Viết chương trình sắp xếp các phần tử của mảng theo thứ tự không giảm.

Hãy gõ, tìm hiểu và chạy thử chương trình sau đây với các giá trị khác nhau của n. Qua đó, rút ra nhận xét về thời gian chạy của chương trình.

```

Program Sorting;
Const Nmax=10000;
Type MyArray=Array[1..Nmax] Of Integer;
Var      A : MyArray;
      n,i,j, k, tg : Integer;
Begin   Randomize;
      Write('Nhap n = '); ReadLn(n);
      {Tạo ngẫu nhiên mảng gồm n số nguyên}
      For i:= 1 to n do A[i]:= Random(300)- Random(300);
      For i:= 1 To n-1 Do
      Begin
        k:= i;    { Tìm chỉ số của phần tử nhỏ nhất trong đoạn từ i đến n }
        For j:= i+1 To n Do
          If A[j] < A[k] Then k:= j;
        { Đổi chỗ A[i] và A[k] }
        tg:= A[i]; A[i]:= A[k]; A[k]:= tg;
      End;
      Write('Da sap xep xong. Go ENTER de xem ket qua...');
      ReadLn;

```

```

        { Đưa kết quả ra màn hình }
    Writeln('Day duoc sap: ');
    For i:= 1 To n Do Write(A[i]:4);
    ReadLn
End.

```

Bài 2

a) Hãy đọc và tìm hiểu những phân tích để viết chương trình giải bài toán sau : Cho mảng A gồm n phần tử. Hãy viết chương trình tạo mảng B[1..n], trong đó B[i] là tổng của i phần tử đầu tiên của A.

Nếu không suy nghĩ kỹ có thể viết chương trình sau đây để giải quyết bài toán đặt ra.

```

Program SubSum1;
Const nmax=100;
Type MyArray=Array[1..nmax] Of Integer;
Var A, B : MyArray;
    n,i,j : Integer;
Begin
    Randomize;
    Write('Nhap n =');
    ReadLn(n);
    {Tạo ngẫu nhiên mảng gồm n số nguyên}
    For i:= 1 to n do A[i]:= Random(300)- Random(300);
    {Bắt đầu}
    For i:= 1 To n Do
        Begin
            B[i]:= 0;
            For j:= 1 to i do B[i]:= B[i]+A[j];
        End;
    {Kết thúc}
    For i:= 1 To n Do Write (B[i]:6);
    ReadLn
End.

```

Để ý rằng ta có hệ thức sau :

$$B[i] = A[1] + A[2] + \dots + A[i]$$

$$B[i] = B[i-1] + A[i], 1 < i \leq n.$$

Phân tích đó cho phép thay đoạn chương trình từ chú giải {Bắt đầu} đến {Kết thúc} bởi đoạn lệnh sau :

$B[1] := A[1];$

For i:= 2 To n Do $B[i] := B[i-1] + A[i];$

b) Bài toán “Mảng con lớn nhất”: Cho mảng số nguyên $a[1..n]$. Mảng $a[p..q]$ với $1 \leq p \leq q \leq n$ được gọi là mảng con của mảng a. Ta gọi trọng lượng của mảng con là tổng các phần tử của nó. Vấn đề đặt ra là trong số các mảng con của a hãy tìm mảng con với trọng lượng lớn nhất. Mảng con như vậy ta sẽ gọi là mảng con lớn nhất. Cần tìm và đưa ra màn hình trọng lượng và chỉ số của phần tử đầu tiên và chỉ số của phần tử cuối cùng của mảng con lớn nhất. Để giải bài toán này cần xét tất cả các mảng con của mảng đã cho. Chương trình có thể cài đặt như sau :

Program MaxSubSeq;

Const

nmax=100;

Type

MyArray=Array[1..nmax] Of Integer;

Var

A : MyArray;

s, max, n, i, j, k, imax, jmax : Integer;

Begin

Write('Nhap n =');

ReadLn(n); {Tạo ngẫu nhiên mảng gồm n số nguyên}

For i:= 1 to n do $A[i] := \text{Random}(300) - \text{Random}(300);$

{Bắt đầu}

max:= maxint; {maxint = 32767 }

For i:= 1 to n do { i : Điểm bắt đầu của mảng con }

for j:= i to n do { j : Điểm kết thúc của mảng con }

begin

s:=0;

for k:=i to j do {Tính trọng lượng của dãy con $a[i..j]$ }

s:=s+a[k];

if s > max then

begin

max:=s; imax:=i; jmax:=j;

end

end;

Writeln('Trong luong cua day con lon nhat : ',max);

Writeln('Day con lon nhat bat dau tu chi so : ',imax);

```
WriteLn('Day con lon nhat ket thuc o chi so : ',jmax); ReadLn ;  
End.
```

Yêu cầu : Hãy tìm cách viết lại chương trình trên trong đó giảm bớt vòng lặp theo k để có được chương trình hiệu quả hơn.

§5. BÀI THỰC HÀNH SỐ 5

Bài 1. Nhập vào từ bàn phím một xâu. Kiểm tra xem nó có phải là một palindrome hay không? (Xâu được gọi là palindrome nếu như đọc nó từ phải sang trái cũng như đọc nó từ trái sang phải).

a) Hãy gõ và chạy thử chương trình sau :

```
Var i,x : byte;  
    a,p : string;  
Begin  
    write('Nhap vao xau : ');  
    readln(a);  
    x:=length(a);           {xác định độ dài của xâu }  
    p:="";                  {khởi tạo xâu rỗng }  
    for i:=x downto 1 do  
        p:=a[i]+p;          {tạo xâu đảo ngược }  
        if a=p then  
            write('Xau la palindrome')  
        else  
            write('Xau khong la palindrome');  
    readln  
end.
```

b) Yêu cầu thêm : Hãy viết lại chương trình trên trong đó không cần có biến xâu p.

Bài 2. Viết chương trình nhập từ bàn phím một xâu kí tự S và thông báo ra màn hình số lần xuất hiện trong S của mỗi chữ cái tiếng Anh (không phân biệt chữ hoa hay chữ thường).

Bài 3. Nhập vào từ bàn phím một xâu. Thay thế tất cả các từ 'anh' trong nó bằng từ 'em'.

§6. BÀI THỰC HÀNH SỐ 6

Bài 1. Xây dựng hàm POWERINT(a,x) tính giá trị của ax trong đó a và x là các số nguyên. Hãy viết chương trình sử dụng hàm đã xây dựng, để tính giá trị của ax với các giá trị của a và x khác nhau. Từ đó, xác định các khoảng giá trị của a và x mà hàm POWERINT(a,x) cho giá trị đúng của biểu thức ax

Bài 2. Tạo hiệu ứng chữ chạy trên màn hình.

a) Trước hết, hãy tìm hiểu việc xây dựng hai thủ tục sau đây :

- Thủ tục CatDan(S1,S2) nhận đầu vào là xâu S1 gồm không quá 79 kí tự, tạo xâu S2 thu được từ xâu S1 bằng việc chuyển kí tự đầu tiên của nó xuống vị trí cuối cùng. Ví dụ nếu S1='abcd' thì S2 = 'bcda'.

```
Type str79 = string[79];
```

```
procedure CatDan(S1: str79; var S2: str79);
```

```
begin
```

```
    s2:= copy(s1,2,length(S1)-1)+s1[1];
```

```
end;
```

- Thủ tục CanGiua(S) nhận đầu vào là xâu S gồm không quá 79 kí tự, bổ sung vào đầu S một số kí tự trắng để khi đưa ra màn hình xâu kí tự trong S ban đầu được căn vào giữa dòng gồm 80 kí tự.

```
procedure CanGiua(Var S: str79);
```

```
    var i, n: integer;
```

```
begin
```

```
    n:= length(S);
```

```
    n:= (80-n) div 2;
```

```
    for i:= 1 to n do S:= ' '+S;
```

```
end;
```

b) Theo dõi cách sử dụng hai thủ tục trên, ta có thể viết chương trình sau đây để viết chương trình nhập một dòng kí tự từ bàn phím và đưa ra dòng chữ chạy ở giữa màn hình văn bản.

```
uses crt;
```

```
Type str79 = string[79];
```

```
var s1, s2: str79;
```

```
procedure CatDan(S1: str79; var S2: str79);
```

```
begin
```



```

    s2:= copy(s1,2,length(S1)-1)+s1[1];
end;
procedure CanGiua(Var S: str79);
var i, n : integer;
begin
    n:= length(S);
    n:= (80-n) div 2;
    for i:= 1 to n do S:= ' '+S;
end;
BEGIN
    clrscr;
    write('Nhap xau S1 : '); readln(s1);
    Cangiuu(S1);
    clrscr;
    repeat
        gotoxy(1,12); (* Chuyển con trỏ đến vị trí đầu dòng 12 *)
        write(S1);
        delay(500); (* Dừng 500 miligiây *)
        CatDan(S1, S2);
        S1:=S2;
    until keypressed; (* Gõ một phím bất kì để kết thúc *)
    readln
END.

```

Hãy chạy thử chương trình trên với dòng chữ

'... Chao SEAGAMES 22 Viet Nam 2003 ... '

c) Hãy viết thủ tục ChuChay(S,Dong) nhận đầu vào là xâu S gồm không quá 79 kí tự và biến nguyên Dong, đưa ra dòng chữ (nội dung xác định bởi S) chạy ở dòng với chỉ số cho bởi giá trị của biến Dong. Viết và chạy chương trình có sử dụng thủ tục này.

§7. BÀI THỰC HÀNH SỐ 7

a) Tìm hiểu việc xây dựng các hàm và thủ tục, thực hiện các việc sau đây có liên quan đến tam giác :

- Tính độ dài các cạnh,
- Tính chu vi,
- Tính diện tích,

- Kiểm tra các tính chất : Đều, Cân, Vuông được trình bày dưới đây.

Giả thiết tam giác được xác định bởi toạ độ của ba đỉnh. Ta sử dụng kiểu dữ liệu sau đây để mô tả một tam giác :

```
Const
    eps = 1.E-6;
Type
    Diem = record
        x, y: real;
    end;
    Tamgiac = record
        A, B, C: Diem;
    end;
```

Ta xây dựng các thủ tục và hàm :

Thủ tục: Procedure DaiCanh(var R : Tamgiac; var a, b, c : real);
nhận đầu vào là biến mô tả tam giác R và đầu ra là độ dài của 3 cạnh a,b,c.

Hàm: Function Chuvi(var R : Tamgiac) : real;
tính chu vi của tam giác R.

Hàm: Function Dientich(var R : Tamgiac) : real;
tính diện tích của tam giác R.

Thủ tục: Procedure Tinhchat(var R : Tamgiac; var Deu, Can, Vuong: boolean);

nhận đầu vào là biến mô tả tam giác R và đầu ra là tính chất của tam giác : Deu hay Can hay Vuong.

Thủ tục: Procedure Show(var R: Tamgiac);
để hiển thị toạ độ ba đỉnh tam giác lên màn hình.

Hàm: Function dist(P, Q : Diem): real;
để tính khoảng cách giữa hai điểm P, Q.

b) Tìm hiểu chương trình nhập vào toạ độ 3 đỉnh một tam giác và sử dụng các hàm, thủ tục được xây dựng dưới đây để khảo sát các đặc tính của tam giác.

```
Uses crt;
```

```
Const
    Eps = 1.E-6;
Type
    Diem = record
        x, y: real;
    end;
    Tamgiac = record
        A, B, C: Diem;
    end;
```

```

Var      T : Tamgiac;
      D, C, V : boolean;
Function dist(P, Q : Diem) : real;
begin
  dist:= sqrt((P.x-Q.x)* (P.x-Q.x)+ (P.y-Q.y)* (P.y-Q.y));
end.
Procedure Daicanh(var R: Tamgiac; var a, b, c: real);
begin
  a:= dist(R.B, R.C);
  b:= dist(R.A, R.C);
  c:= dist(R.A, R.B);
end;
Function ChuVi(var R: Tamgiac): real;
var a, b, c: real;
begin
  Daicanh(R, a, b, c);
  ChuVi:= a + b + c;
end;

Function Dientich(var R: Tamgiac): real;
var a, b, c, p: real;
begin
  Daicanh(R, a, b, c);
  p:= (a+b+c)/2;
  Dientich:= sqrt(p*(p-a)*(p-b)*(p-c))
end;
Procedure Show(var R: tamgiac);
begin
  writeln('Toa do 3 dinh cua tam giac la:');
  writeln(' Dinh A',R.A.x:0:3, R.A.y:0:3);
  writeln(' Dinh B',R.B.x:0:3,R.B.y:0:3);
  writeln(' Dinh C',R.C.x:0:3,R.C.y:0:3);
end;
Procedure Tinhchat(var R: Tamgiac; var Deu,Can,Vuong: boolean);
var a,b, c: real;
begin
  Deu:= false; Can:= false; Vuong:= false;
  Daicanh(R, a, b, c);
  if (abs(a-b)<eps) and (abs(a-c)<eps) then
    begin
      Deu:= true;
      can:= true;

```

```

    end
  else
    if (abs(a-b)<eps) or (abs(a-c)<eps) or (abs(b-c)<eps) *
    then Can:= true;
    if (abs(a*a+b*b-c*c)<eps) or (abs(a*a+c*c-b*b)<eps)
      or (abs(b*b+c*c-a*a)<eps) then Vuong:= true;
    end;
  BEGIN {chương trình chính}
    Writeln('Nhap tam giac :');
    Write('Toa do dinh A :'); readln(T.A.x, T.A.y);
    Write('Toa do dinh B :'); readln(T.B.x, T.B.y);
    Write('Toa do dinh C :'); readln(T.C.x, T.C.y);
    Writeln('=====');
    Show(T);
    Writeln('Dien tich',Dientich(T):9:3);
    Writeln('Chu vi :',Chuvi(T):9:3);
    Tinhchat(T, D, C, V);
    Writeln('Tam giac co tinh chat :');
    If D then writeln(' la tam giac deu')
    Else if C then writeln(' la tam giac can');
    If V then writeln(' la tam giac vuong');
    Readln;
  END.

```

END.

c) Viết chương trình sử dụng các hàm và thủ tục xây dựng ở trên để giải bài toán sau : Cho tệp dữ liệu TAMGIAC.DAT có cấu trúc như sau :

- Dòng đầu tiên chứa số N
- Mỗi một trong số N dòng tiếp theo chứa 6 số thực $x_A, y_A, x_B, y_B, x_C, y_C$ là các toạ độ của ba đỉnh $A(x_A, y_A), B(x_B, y_B), C(x_C, y_C)$ của một tam giác ABC.

Hãy nhập dữ liệu từ tệp đã cho và đưa ra màn hình :

- Số lượng tam giác đều;
- Số lượng tam giác cân (nhưng không là đều);
- Số lượng tam giác vuông trong số N tam giác đã cho.

§8. BÀI THỰC HÀNH SỐ 8

Bài 1. Viết hàm IsPalin(n) nhận đầu vào là số nguyên dương n không vượt quá 2147483647, có giá trị True nếu n là số palindrome và giá trị False nếu trái lại. (Một số nguyên dương được gọi là số palindrome nếu như đọc từ trái qua phải cũng như đọc từ phải qua trái ta có cùng một số).

Bài 2. Sử dụng hàm trong câu a xây dựng chương trình giải bài toán sau đây. Cho tệp văn bản DAYSO.INP có cấu trúc sau :

- Dòng đầu tiên chứa số nguyên dương N ($N < 1000$);
- Dòng thứ i trong số N dòng tiếp theo chứa số nguyên dương a_i , $i = 1, 2, \dots, N$.

Hãy nhập các phần tử của dãy số $a_1, a_2, \dots, a_N$ từ tệp văn bản đã cho. Xác định xem trong dãy số đã cho có bao nhiêu số là palindrome. Kết quả đưa ra tệp văn bản có tên DAYSO.OUT theo khuôn dạng sau :

- Dòng đầu tiên ghi K là số lượng số palindrome;
- Mỗi dòng trong số K dòng tiếp theo chứa chỉ số của một số hạng của dãy số là số palindrome.

Ví dụ

Các tệp dữ liệu vào/ra có thể có dạng :

DAYSO.INP	DAYSO.OUT
3	2
131	1
1234	3
12344321	

C. MỘT SỐ ĐỀ KIỂM TRA THAM KHẢO

(Thời gian 45 phút).

ĐỀ 01.

I. Phần trắc nghiệm: (3 điểm)

- Bài toán có giải thuật là bài toán:
 - Có lời giải và có kết quả sau một số bước hữu hạn.
 - Có kết quả.
 - Có các bước giải.
 - Có lời giải hay.
- Ngôn ngữ lập trình Pascal là:
 - Ngôn ngữ lập trình bậc thấp.
 - Ngôn ngữ máy tính.
 - Ngôn ngữ lập trình bậc cao.
 - Ngôn ngữ biên dịch.
- Trong một ngôn ngữ lập trình cú pháp dùng để?
 - Biên soạn chương trình.
 - Biên dịch chương trình.
 - Xác định các thao tác thực hiện.
 - Làm quy tắc viết chương trình.
- Trong ngôn ngữ lập trình Pascal để chú thích một nội dung ta sử dụng:
 - Cặp dấu { và }.
 - Cặp dấu /* và */.
 - Cặp dấu (* và *).
 - Cả hai đáp án A và C đều đúng.

II. Phần tự luận: (7 điểm)

Câu 1: Viết chương trình giải phương trình $ax^2 + bx + c = 0$.

Câu 2: Viết chương trình nhập một mảng một chiều gồm 20 phần tử, tính tổng giá trị các phần tử có giá trị chẵn và dương, hiển thị kết quả ra màn hình.

ĐÁP ÁN ĐỀ 01.

I. Phần trắc nghiệm: 1.A. 2.C. 3.D. 4.D.

II. Phần tự luận:

Câu 1: (2 điểm)

```
var a, b, c, dt, x1, x2:real;
Begin
write('Nhap ba so bat ky');
readln(a,b,c);
if a=0 then
    begin
        if b=0 then
            Begin
                if c=0 then write('PTVSN');
                if c<>0 then write('PTVN');
            End;
        If b<>0 then write('PTCN x=', -c/b:8:2)
        End;
    if a<>0 then
        Begin
            dt:=b*b-4*a*c;
            if dt<0 then write('PTVN');
            if dt=0 then write('PTCN x=', -b/(2*a):8:2);
            if dt>0 then
                begin
                    x1:=(-b+sqrt(dt))/(2*a);
                    x2:=(-b-sqrt(dt))/(2*a);
                    writeln('PT co hai nghiệm phân biệt');
                    write(x1:8:2, ' ', x2:8:2);
                end;
        End;
    End;
Readln;
```

End.

Câu 2: (5 điểm)

Type mmc=array[1..20] of longint;

Var a:mmc; s:longint; i:byte;

Begin

For i:=1 to 20 do

```

begin
write('a[' ,i,']='); readln(a[i]);
End;
s:=0;
For i:=1 to 20 do
    if (a[i] mod 2=0) and (a[i]>0) then s:=s+a[i];
write('Tong la ',S);
readln;
End.

```

ĐỀ SỐ 02.

I. Phần trắc nghiệm: (3 điểm)

1. Trong ngôn ngữ lập trình Tên là:
 - A. Cách gọi của các giá trị.
 - B. Một dãy liên tiếp các kí tự có độ dài không quá 255 kí tự bao gồm các chữ số, chữ cái hoặc dấu gạch dưới và bắt đầu bằng chữ cái hoặc dấu gạch dưới.
 - C. Một dãy liên tiếp các kí tự có độ dài không quá 127 kí tự bao gồm các chữ số.
 - D. Một dãy liên tiếp các kí tự có độ dài bất kì bao gồm các chữ số, chữ cái hoặc dấu gạch dưới và bắt đầu bằng chữ cái hoặc dấu gạch dưới.
2. Trong các biểu diễn hằng dưới đây biểu diễn nào đúng?
 - A. True; 123.456; A21.
 - B. False; 'Pascal'; 3.14.
 - C. 3.14; E1.6E-5; '43'.
 - D. -25; "hoc bai"; true.
3. Trong các phát biểu sau phát biểu nào sai:
 - A. Trong một chương trình phần khai báo có thể có hoặc không.
 - B. Trong một chương trình phần thân chương trình có thể có hoặc không.
 - C. Trong một chương trình phần thân chương trình bắt buộc phải có.
 - D. Trong một chương trình ít nhất phải có một phần là phần thân chương trình.
4. Cú pháp để khai báo hằng trong chương trình Pascal là:
 - A. Const <tên hằng>.
 - B. Const <Giá trị của hằng>.
 - C. Const <tên hằng> = <Giá trị>.
 - D. <tên hằng> = <Giá trị>.

II. Phần tự luận: (7 điểm)

Câu 1: Hãy phân biệt kỹ thuật biên dịch và thông dịch.

Câu 2: So sánh sự giống và khác nhau của cấu trúc For và While.

Cho chương trình có sử dụng cấu trúc For như sau:

```

var i:byte
Begin
    For i:=1 to 30 do write(i:4);
    Readln;
End.

```

Hãy viết lại chương trình bằng cách thay cấu trúc For bằng cấu trúc While.

Câu 3: Viết chương trình nhập một mảng một chiều gồm 20 phần tử, đếm số phần tử có giá trị âm.

ĐÁP ÁN ĐỀ 02.

I. Phần trắc nghiệm: 1.B. 2.B. 3.B. 4.C.

II. Phần tự luận:

Câu 1: (1 điểm)

Biên dịch:

Bước 1 : Duyệt, phát hiện lỗi, kiểm tra tính đúng đắn của lệnh trong chương trình nguồn.

Bước 2 : Dịch toàn bộ chương trình nguồn thành một chương trình trên ngôn ngữ máy.

(thuận tiện cho các chương trình ôn định và cần thực hiện nhiều lần)

- Thông dịch:

Bước 1 : Kiểm tra tính đúng đắn của lệnh tiếp theo trong chương trình nguồn.

Bước 2 : Chuyển lệnh đó thành ngôn ngữ máy.

Bước 3 : Thực hiện các câu lệnh vừa được chuyển đổi.

(phù hợp với môi trường đối thoại giữa người và máy)

Câu 2: (2 điểm)

- Giống: For và While đều cùng là cấu trúc lặp

- Khác: For là cấu trúc lặp có số lần đã biết trước, ngược lại While là cấu trúc lặp có số lần chưa xác định.

```

var i:byte
Begin
    i:=1;
    While i<=30 do
        begin
            write(i:4);
            i:=i+1;
        End;
    Readln;
End.

```

Câu 3: (4 điểm)

Type mmc=array[1..20] of longint;

Var a:mmc; s:byte; i:byte;

```

Begin
For i:=1 to 20 do
    begin
        write('a[',i,']='); readln(a[i]);
    End;
s:=0;
For i:=1 to 20 do
    if a[i]<0 then s:=s+1;
write('So luong dem duoc la ',S);
readln;
end.

```

ĐỀ 03.

I. Phần trắc nghiệm (3 điểm)

- i. Trong các khai báo sau khai báo nào sai?
 - A. Program bai1; Uses Crt; Const 3.14;
 - B. Const giatri = 123.56; Program baihocdautien;
 - C. Program Vi_du; Const pi = 3.14; USES CRT;
 - D. Program VD; Const a=12; Const = '123'
2. Kiểu dữ liệu Integer có phạm vi giá trị là:
 - A. Từ 2^{15} đến 2^{16} .
 - B. Từ -2^{15} đến 2^{16} .
 - C. Từ -2^{15} đến $2^{15}-1$.
 - D. Từ -2^{15} đến $2^{16}-1$.
3. Kiểu dữ liệu Real có phạm vi giá trị là:
 - A. 0
 - B. Giá trị tuyệt đối trong phạm vi từ 10^{-38} đến 10^{39} .
 - C. Giá trị tuyệt đối trong phạm vi từ 10^{-38} đến $10^{39}-1$.
 - D. 0 hoặc giá trị tuyệt đối trong phạm vi từ 10^{-38} đến 10^{38} .
4. Cú pháp để khai báo biến là:
 - A. Var <danh sách biến>;
 - B. <Danh sách biến>: Kiểu dữ liệu;
 - C. Var <Danh sách biến>: <Kiểu dữ liệu>;
 - D. Var <Danh sách biến> = <Kiểu dữ liệu>;

II. Phần tự luận: (7 điểm)

Câu 1: Hãy so sánh sự giống và khác nhau của kiểu mảng một chiều và kiểu bản ghi.

Câu 2: Cho chương trình như sau:

```

Program cau 2;
Type mmc=array[1..300] of integer;
Var f:text; a:mmc;

```

Begin

```
assign(f, 'cau2.inp');  
reset();  
readln(f,n);  
for i:=1 to n do read(f,a[i]);  
close(f);  
max:=a[1];  
for i := 2 to n do  
    if max<a[i] then max:=a[i];  
write(f,max);  
Readln;
```

End.

Yêu cầu:

- Tìm lỗi cú pháp và sửa để khi biên dịch chương trình sẽ không còn lỗi.
- Cho file cau2.inp có cấu trúc và nội dung như sau:

Dòng 1: Ghi số 6.

Dòng 2: Ghi 6 số: 7 3 6 8 21 2

Hỏi kết quả sau khi thực hiện chương trình bằng bao nhiêu?

Câu 3. Viết chương trình dưới dạng chương trình con tìm ước chung lớn nhất và bội chung nhỏ nhất của hai số.

ĐÁP ÁN ĐỀ 03.

I. Phần trắc nghiệm: 1.A. 2. C. 3D. 4C.

II. Phần tự luận:

Câu 1: (1 điểm)

- Mảng một chiều và bản ghi đều được ghép từ nhiều phần tử lại với nhau
- Mảng một chiều: Các phần tử phải cùng kiểu dữ liệu, các phần tử phân biệt nhau bằng chỉ số.
- Bản ghi: Các phần tử có kiểu dữ liệu có thể khác nhau, các phần tử phân biệt nhau bởi tên trường.
- Chính vì những đặc điểm đó nên dẫn đến khác nhau trong tham chiếu đến từng phần tử của hai biến thuộc hai kiểu dữ liệu này.

Câu 2: (2 điểm)

- Sửa lỗi cú pháp:

Program cau 2;

Type mmc=array[1..300] of integer;

Var f: text; a:mmc; i,n:word;

Begin

```
assign(f, 'cau2.inp');  
reset(f);  
readln(f,n);
```



```

for i:=1 to n do read(f,a[i]);
close(f);
max:=a[1];
for i:=2 to n do
    if max<a[i] then max:=a[i];
write(max);
Readln;

```

End.

- Kết quả in ra màn hình là 21.

Câu 3: (4 điểm)

Procedure nhap;

```

Begin
Writeln('Nhap vao hai so nguyen duong');
Readln(a,b);
End;

```

Function uscln(a,b:word):Word;

```

Begin
While a<>b do
    if a>b then a:=a-b else b:=b-a;
uscln:=a;
End;

```

Begin

Nhap;

bscnn:=uscln(a,b)/(a*b);

writeln(bscnn);

readln;

End.

ĐỀ 04.

I. Phần trắc nghiệm: (3 điểm)

1. Kiểu dữ liệu Longint có phạm vi giá trị là:

- A. Từ -2^{31} đến $2^{31} - 1$.
- B. Từ -2^{31} đến 2^{31} .
- C. Từ 2^{31} đến $2^{31} - 1$.
- D. Từ -2^{31} đến $2^{31} + 1$.

2. Trong khai báo biến, nếu có nhiều biến có cùng một kiểu thì giữa các biến cách nhau bởi:

- A. Dấu chấm (.).
- B. Dấu chấm phẩy (;).
- C. Dấu phẩy (,).
- D. Dấu hai chấm (:).

3. Trong khai báo biến:

Var Ch: Char;
 A, B: Integer;
 C, D: Byte;
 E, F: Real;

tổng bộ nhớ dành cho các biến đã khai báo là:

- A. 19. B. 21. C. 25 D. 29.

4. Trong khai báo biến:

Var Ch: Char;
 A, B: Integer;
 C, D: Byte;
 E, F: Boolean;

Các giá trị mà biến nhận được dưới đây giá trị nào đúng?

- A. Ch = 'A'; a = 5; b = 6; c = 8; d = 10; e = 1; f = 2.
 B. Ch = 'B'; a = 0; b = 3; c = 6; d = 10; e = 1; f = True.
 C. Ch = 'O'; a = 0; b = 3; c = 6; d = 10; e = 1; f = False.
 D. Ch = '1'; a = 2; b = 4; c = 6; d = 10; e = True; f = False.

II. Phần tự luận: (7 điểm)

Câu 1. Viết chương trình để in ra màn hình bảng cửu chương có dạng như sau:

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

Câu 2. Cho biết chương trình dưới đây làm gì?

```
Program Sum1;
Uses Crt;
Const nmax:=100;
Type Myarray = Array[1..nmax] of integer;
Var A:myarray;
    s, n, i, k:integer;
Begin
  Clrscr;
  Randomize;
  Write('Nhap n=');
  Readln(n);
```

```

For i:=1 to n do a[i]:=random(300)-random(300);
For i:=1 to n do Write(A[i]:5);
Writeln;
Write('Nhap k=');
Readln(k);
s:=0;
For i:=1 to n do
    if a[i] mod k=0 then s:=s+a[i];
Write('Tong can tinh la ', s);
Readln;
End.

```

Câu 3. Thêm các lệnh mới vào chương trình nhằm sửa đổi chương trình trong câu 2 để chương trình thực hiện đếm số lượng số dương và số lượng số âm của mảng.

```

Posi, neg: integer;
Posi:=0;neg:=0;
If a[i]>0 then Posi:=posi+1
Else if a[i]<0 then neg:=neg+1;
Write(posi:4,neg:4);

```

ĐÁP ÁN ĐỀ 04.

I. Phần trắc nghiệm: 1.A 2. C. 3.A 4. D

II. Phần tự luận:

- Nhập chương trình và xử lí.

ĐỀ 05.

I. Phần trắc nghiệm: (3 điểm)

1. Trong Pascal các biểu diễn nào dưới đây là đúng?

- A. $5a - 4c$; $(12*a + 5b)/2$; $(x-y)*z$.
- B. $\text{delta} = b^2 - 4ac$; $x_1 = x_2 = -b/2a$;
- C. $1*a*(1-a)+2$; $13*a - (1/x) + (x-y)*(y-x)$.
- D. $(a-b)/(c-d) + (a-1/c)$; $2x^2 - 4x + 2$.

2. Cho các khai báo

```

Var    Ch: Char;
       A, B: Integer;
       C, D: Byte;
       E, F: Real;

```

Các phép gán nào dưới đây là đúng?

- A. $A:=b$; $b:=c$; $d:=f$;
- B. $c:=d$; $e:=f$; $ch:=c$;

C. A:=b; c:=d; e:=f; ch:='a'; c:=f;

D. A:=2; b:=a; e:=3.14; f:=e; ch:='a';

3. Cú pháp để nhập dữ liệu vào từ bàn phím là:

A. Read(<danh sách biến vào>);

B. Readln(<danh sách biến vào>);

C. Readln;

D. Đáp án A và B đều đúng.

4. Kết quả của đoạn chương trình sau là gì?

```
Var    i:byte;
```

```
Begin
```

```
      I:= 12345;
```

```
      Writeln(I);
```

```
      Write(-456);
```

```
End.
```

A. 12345; -456.

B. 12345

 -456

C. -456; 12345.

D. -456

 12345.

II. Phần tự luận: (7 điểm)

Câu 1. Viết chương trình nhập một mảng một chiều A[1..20] và nhập một số x. Đếm số lượng số trong A có giá trị bằng x.

Câu 2: Viết chương trình sắp xếp các phần tử của mảng theo thứ tự không giảm.

ĐÁP ÁN ĐỀ 05.

I. Phần trắc nghiệm: 1. C. 2. D 3D. 4. B

II. Phần tự luận:

Câu 1. Tham khảo giải thuật bài 11. (3 điểm)

Câu 2. Chương trình minh họa: (4 điểm)

```
Uses CRT;
```

```
Const    nmax=250;
```

```
Type    arrint=array[1..nmax] of integer;
```

```
Var    n, i, j, y:integer;
```

```
      a:arrint;
```

```
Begin
```

```
      clrscr;
```

```
      Randomize;
```

```

Write('Nhap n=');
Readln(n);
For i:=1 to n do a[i]:=random(300)-random(300);
For i:=1 to n do write(a[i]:5);
Writeln;
For j:=n downto 2 do
    For i:=1 to j-1 do
        If a[i]>a[i+1] then
            Begin
                t:=a[i];
                a[i]:=a[i+1];
                a[i+1]:=t;
            End;
Writeln('Day so sau khi sap xep');
For i:=1 to n do write(a[i]:7);
Writeln;
Readln;
End.

```

ĐỀ 06.

I. Phần trắc nghiệm: (3 điểm)

1. Kết quả của đoạn chương trình sau là gì?

```

Var i:byte;
Begin
    I:= 12345;
    Writeln(I);
    Write(-456);
End.

```

- A. 12345; -456.
- B. 12345
-456
- C. -456; 12345.
- D. -456
12345.

2. Để chạy chương trình ta sử dụng tổ hợp phím nào?

- A. Ctrl + F4.
- B. Ctrl + F6.

C. Ctrl + F9.

D. Ctrl + F12.

3. Cú pháp biểu diễn cấu trúc rẽ nhánh dạng thiếu là:

A. If <biểu thức> then < câu lệnh>

B. If <biểu thức> then < câu lệnh> else <câu lệnh>

C. If <biểu thức điều kiện> then < câu lệnh>

D. If <biểu thức điều kiện> then < câu lệnh> else <câu lệnh>

4. Kết quả của đoạn chương trình sau là gì?

Var x: integer;

Begin

X:= 10;

If (x - 3) > 5 then

Writeln(' xin chuc mung');

End.

A. Không hiển thị gì cả.

B. In ra kết quả là 7.

C. In ra màn hình câu: 'xin chuc mung'.

D. In ra kết quả là 5.

II. Phần tự luận: (7 điểm)

Câu 1. Cho mảng A gồm n phần tử. Viết chương trình tạo mảng B[1..n], trong đó B[i] là tổng giá trị của i phần tử đầu tiên của mảng A.

Câu 2. Viết chương trình nhập vào một xâu bất kì, in ra màn hình xâu đó sau khi đã xoá bỏ hết các kí tự trắng dư thừa, trong đó các kí tự trắng dư thừa là các kí tự trắng đứng ở đầu xâu, cuối xâu hoặc hai kí tự trắng liền nhau.

ĐÁP ÁN ĐỀ 06.

I. Phần trắc nghiệm: 1. B. 2. C. 3. C. 4. C

II. Phần tự luận:

Câu 1. (4 điểm)

Program xauchuan;

Var st:string;

i:integer;

Begin

Writeln('Nhap vao mot chuoai bat ky');

Readln(st);

While st[1]=' ' do

delete(st,1,1);

While st[length(st)]=' ' do

```

delete(st,length(st),1);
For i:= 1 to length(st) -1 do
While (st[i]=' ') and (st[i+1]=' ') do
delete(st,i,1);
Write('Xau chuan la ',st);
Readln;

```

End.

Đâu 2. (3 điểm).

Chương trình minh họa:

```

Const nmax=100;
Type myarray=array[1..nmax] of integer;
Var n, i, j:integer;
    a,b:arrint;

Begin
Rardomize;
Write('Nhap n=');
Readln(n);
For i:=1 to n do a[i]:=random(300)-random(300);
For i:=1 to n do write(a[i]:5);
Writeln;
For i:=1 to n do
    Begin
    B[i]:=0;
    For j:=1 to i do B[i]:=B[i]+A[j];
    End;
For i:=1 to n do write(B[i]:5);
Readln;
End

```

ĐỀ 07.

Phần trắc nghiệm: (2 điểm)

- . Kết quả của việc thực hiện hàm concat('abc', 'de', '123') cho kết quả là:
 - A. 'abcde'.
 - B. '123abcde'.
 - C. 'abcde123'.
 - D. 'abc123'.
- . Cấu trúc lặp với số lần biết trước là cấu trúc:
 - A. Repeat Until.
 - B. While Do.
 - C. For to do.
 - D. For do.
- . Trong vòng lặp While <BTĐK> do <câu lệnh>, phát biểu nào dưới đây đúng?
 - A. Chừng nào biểu thức điều kiện còn sai thì thực hiện câu lệnh.
 - B. Trong vòng lặp các câu lệnh thực hiện ít nhất là 1 lần.
 - C. Chừng nào biểu thức điều kiện còn đúng thì thực hiện câu lệnh.

D. Các câu lệnh trong vòng lặp không phải nằm giữa từ khoá Begin...End;

II. Phần tự luận: (8 điểm)

Câu 1: (3 điểm)

Viết chương trình nhập vào một mảng n số nguyên. Sau đó thực hiện các công việc:

- + Kiểm tra n thoả mãn lớn hơn 0 và nhỏ hơn 20 thì mới nhập số phần tử cho mảng.
- + Nhập vào một giá trị x nguyên bất kì và đếm xem trong mảng nói trên có bao nhiêu phần tử bằng x, in ra màn hình số phần tử đó.

Câu 2: (5 điểm)

Viết chương trình quản lí xe máy, biết rằng mỗi xe máy được quản lí bởi các thông tin sau: Tên xe, nước sản xuất, giá tiền. Thực hiện các công việc:

- + Nhập vào danh sách các loại xe máy
- + Hiện thị ra màn hình các loại xe máy đã nhập
- + Tìm và hiện thị ra màn hình thông tin danh sách các loại xe máy có nguồn gốc từ nước Trung Quốc.

ĐÁP ÁN ĐỀ 07.

I. Phần trắc nghiệm: 1. C. 2. C. 3. C.

II. Phần tự luận:

Câu 1.

Program Timphantu;

Var dem,n,i,x: Integer;

A: Array[1..20] of Integer;

Begin

Repeat

Writeln('Nhap vao n');

Readln(n);

Until (n>0) and (n<=30);

dem:=0;

Writeln('Nhap vao gia tri so sanh');

Readln(x);

Writeln('Nhap ma tran A');

For i:= 1 to n do

Begin

Write('a[' ,i,']=');

Readln(a[i]);

End;

For i:= 1 to n do

Begin If a[i]=x then

```

        dem:=dem+1;
    end;
    Writeln('so phan tu cua mang co gia-tri bang x la ',dem);
    Readln;
End.
Câu 2.
Program QUANLYXEMAY;
Uses Crt;
Type XEMAY = Record
    Ten:string;
    nuocsx: string;
    giatien:real;
End;
Var a : array[1..20] of XEMAY;
    i,n:Integer;
Begin
    Clrscr;
    Writeln('Nhap so xe may ');
    Readln(n);
    For i:= 1 to n do
        Begin
            Writeln(' Nhap cac thong tin ve xe may ',i);
            With a[i] do
                Begin
                    Write(' Nhap ten: '); Readln(ten);
                    Write(' Nhap nuoc san xuat: '); Readln(nuocsx);
                    Write(' Nhap gia tien: '); Readln(giatien);
                End;
            End;
        End;
    Readln;
    Writeln('Thong tin ve cac loai xe');
    For i:= 1 to n do
        Begin
            With a[i] do
                Begin
                    Writeln('Ten xe: ',ten);
                    Writeln('Nuoc san xuat: ',nuocsx);
                    Writeln('Gia thanh ',giatien:2:2);
                    Writeln('Thue gia tang ',5*giatien/100:2:2);
                End;
            Readln;
        End;
    End;

```

```

Writeln('Danh sach cac xe co nguon goc tu Trung Quoc');
For i:= 1 to n do
Begin
  With a[i] do
    if nuocsx='Trung Quoc' then
    Begin
      Writeln('Ten xe: ',ten);
      Writeln('Nuoc san xuat: ',nuocsx);
      Writeln('Gia thanh ',giatien:2:2);
    End;
  Readln;
End;
Enc.

```

ĐỀ 08.

I. Phần trắc nghiệm: (2 điểm)

1. Mảng là kiểu dữ liệu:
 - A. Phức tạp.
 - B. Đa cấu trúc.
 - C. Cấu trúc
 - D. Đơn giản.
2. Cú pháp để khai báo biến mảng một chiều gián tiếp là:
 - A. Type <tên biến mảng> : array[kiểu chỉ số] of <Kiểu phần tử>;
Var : Kiểu biến mảng;
 - B. Type <tên kiểu mảng> = [kiểu chỉ số] of <kiểu phần tử>;
Var <tên biến mảng>: <tên kiểu mảng>;
 - C. Type <tên kiểu mảng> = [kiểu phần tử] of <kiểu chỉ số>;
Var <tên biến mảng>: <tên kiểu mảng>;
 - D. Var <tên biến mảng>: <tên kiểu mảng>;
3. Chương trình dịch có mấy loại:
 - A. 1 loại (Biên dịch).
 - B. 2 loại (Thông dịch và biên dịch).
 - C. 3 loại (Thông dịch, biên dịch, chương trình gốc).
 - D. 4 loại (Thông dịch, biên dịch, chương trình gốc, chương trình đích).

II. Phần tự luận: (8 điểm)

Câu 1: (3 điểm)

Viết chương trình nhập vào hai ma trận vuông cấp n với các phần tử của ma trận là số nguyên sau đó tính tổng hai ma trận và tích của hai ma trận đó. Hiện thị kết quả ra màn hình.

Câu 2: (5 điểm)

Để quản lí điểm của học sinh một trường đã lưu các thông tin về mỗi học sinh như sau: số báo danh, tên, điểm toán, điểm lí, điểm hoá. Viết chương trình thực hiện các công việc sau:

- + Nhập vào một danh sách học sinh từ bàn phím.
- + Hiện thị danh sách học sinh ra màn hình kèm theo tổng điểm các môn học.
- + Tìm và in ra danh sách học sinh giỏi. Điều kiện xếp loại học sinh giỏi là điểm trung bình ≥ 8 , biết rằng các môn học có hệ số 1.

ĐÁP ÁN ĐỀ 08.

I. Phần trắc nghiệm: 1.C; 2. B; 3. B

II. Phần tự luận:

Câu 1.

Program Matran;

Var n,i,j,k: Integer;

a,b,c,d: Array[1..10,1..10] of Integer;

Begin

Repeat

Writeln('Nhap vao n');

Readln(n);

Until (n>0) and (n<=10);

Writeln('Nhap ma tran A');

For i:= 1 to n do

For j:= 1 to n do

Begin

Write('a[' ,i ,',' ,j ,'] =');

Readln(a[i,j]);

End;

Writeln('Nhap ma tran B');

For i:= 1 to n do

For j:= 1 to n do

Begin

Write('b[' ,i ,',' ,j ,'] =');

Readln(b[i,j]);

End;

For i:= 1 to n do

For j:= 1 to n do

Begin

c[i,j]:=a[i,j]+b[i,j];

d[i,j]:=0;

For k:= 1 to n do

d[i,j]:=d[i,j]+a[i,k]*b[k,j];

End;

```

Writeln(' Ma tran tong la');
For i:= 1 to n do
Begin
  For j:= 1 to n do Write(c[i,j]:4);
  Writeln;
end;
Writeln(' Ma tran tich la');
For i:= 1 to n do
Begin
  For j:= 1 to n do Write(d[i,j]:4);
  Writeln;
end;
Readln;
End.

```

Câu 2.

```

Program Baithi;
Uses Crt;
Type Hocsinh = Record
    Ten:string;
    Sbd:Integer;
    Toan,ly,hoa:real;
End;
Var a : array[1..20] of hocsinh;
    i,n:Integer;
Begin
  Clrscr;
  Writeln('Nhap so hoc sinh');
  Readln(n);
  For i:= 1 to n do
  Begin
    Writeln(' Nhap thong tin cho hoc sinh thu ',i);
    With a[i] do
    Begin
      Write(' Nhap ten: '); Readln(ten);
      Write(' Nhap so bao danh: '); Readln(sbd);
      Write(' Nhap diem toan: '); Readln(toan);
      Write(' Nhap diem ly: '); Readln(ly);
      Write(' Nhap diem hoa: '); Readln(hoa);
    End;
  End;
  Readln;

```

```

Writeln('Danh sach hoc sinh');
For i:= 1 to n do
Begin
  With a[i] do
  Begin
    Writeln('Ten hoc sinh: ',ten);
    Writeln('So bao danh: ',sbd);
    Writeln('Diem toan ',toan:2:2);
    Writeln('Diem ly ',ly:2:2);
    Writeln('Diem hoa ',hoa:2:2);
    Writeln('Diem trung binh ',(toan+ly+hoa)/3:2:2);
  End;
End;
Read n;
End;
Writeln('Danh sach hoc sinh gioi');
For i:= 1 to n do
Begin
  With a[i] do
    if (toan+ly+hoa)/3>=8.0 then
    Begin
      Writeln('Thong tin ve hoc sinh thu: ',i);
      Writeln('Ten hoc sinh: ',ten);
      Writeln('So bao danh: ',sbd);
      Writeln('Diem toan ',toan:2:2);
      Writeln('Diem ly ',ly:2:2);
      Writeln('Diem hoa ',hoa:2:2);
      Writeln('Diem trung binh ',(toan+ly+hoa)/3:2:2);
    End;
  End;
End;
End.

```

ĐỀ 09.

I. Phần trắc nghiệm: (3 điểm)

1. Trong các khai báo dưới đây khai báo nào đúng:
 - A. Arrayreal = array[10..20] of real;
 - B. Arrayinteger = array[1..100] of integer;
 - C. Logic = array[1..9] of boolean;
 - D. Tất cả đều đúng.
2. Cú pháp nào dưới đây dùng để khai báo gián tiếp biến mảng hai chiều?
 - A. Type <tên biến mảng> = array[Kiểu chỉ số] of <kiểu phần tử>.
 - Var <tên biến mảng> : <tên kiểu mảng>.

B. Type <tên kiểu mảng> = array[Chỉ số dòng, chỉ số cột] of <kiểu phần tử>.

C. Type <tên kiểu mảng> = array[Chỉ số dòng, chỉ số cột] of <kiểu phần tử>.

Var <tên biến mảng> : <tên kiểu mảng>.

D. Type <tên biến mảng> = array[Chỉ số dòng, chỉ số cột] of <kiểu phần tử>.

Var <tên biến mảng> : <tên kiểu mảng>.

3. Giả sử đã khai báo biến mảng và các biến khác, đoạn chương trình dưới đây làm gì?

For i := 1 to m do

Begin

Writeln('a[', i, ']=', a[i]);

End;

A. Tính tổng các phần tử của mảng A.

B. In ra chỉ số các phần tử của mảng.

C. Hiện thị giá trị các phần tử của mảng, mỗi phần tử nằm trên một dòng.

D. Tìm kiếm một phần tử trong mảng.

II. Phần tự luận: (7 điểm)

Câu 1: (3 điểm)

Một chuỗi được gọi là chuỗi Palindrom nếu nó trùng với chuỗi đảo của nó. Viết chương trình nhập vào một chuỗi bất kỳ và cho biết nó có phải là chuỗi Palindrom hay không?

Câu 2: (4 điểm)

Viết chương trình nhập vào một mảng một chiều các số nguyên. Sau đó bổ sung một phần tử vào vị trí thứ k và có giá trị là x. In ra màn hình danh sách sau khi đã bổ sung.

ĐÁP ÁN ĐỀ 09.

I. Phần trắc nghiệm: 1.D; 2.C 3.C

II. Phần tự luận:

Câu 1.

Program chuoiPalindrom;

Var st, st1: string;

i:byte;

Begin

Writeln('Nhap vao mot chuoi bat ki');

Readln(st);

st1:=st;

For i:= length(st) downto 1 do

```

        st1:= st1+st[i];
    If st = st1 then Writeln(st,' la chuoì Palindrom')
    Else Writeln(st,' không phải là chuoì Palindrom');
    Readln;

```

End.

Câu 2.

Program chenphantu;

Uses Crt;

Var A: array[1..20] of integer;

 i,k,x,n: integer;

Begin

 Writeln(' Nhập số phần tử cho mảng');

 Readln(n);

 For i=1 to n do

 Begin

 Write('A[' ,i,']=');

 Readln(a[i]);

 End;

 For i=1 to n do

 Begin

 Writeln(A[i]);

 End;

 Writeln('Nhập vị trí cần chen');

 Readln(k);

 Write(' Nhập giá trị cần chen');

 Readln(x);

 For i:= n+1 downto k+1 do

 a[i]:=a[i-1];

 a[k]:=x;

 Writeln('Kết quả sau khi chen');

 For i:=1 to n+1 do

 Begin

 Writeln(A[i]);

 End;

 Readln;

End.

ĐỀ 10.

I. Phần trắc nghiệm: (3 điểm)

1. Cho đoạn chương trình:

```
Var S:array[1..10] of Integer;
```

```
l, tong: Integer;
```


Begin

```
For i:= 1 to 10 do  
S[i]:= i;  
Tong:=0;  
For i:= 1 to 10 do  
If s[i] mod 2 = 0 then  
Tong:= tong+S[i];  
Write(tong);  
Readln;
```

End.

Khi thực hiện chương trình cho kết quả nào dưới đây?

- A. 30.
- B. 40
- C. 50
- D. 60

2. Xâu là kiểu dữ liệu:

- A. Đa trị.
- B. Có cấu trúc.
- C. Đơn giản chuẩn.
- D. Vô hướng

3. Để định nghĩa một kiểu xâu ta sử dụng cú pháp nào?

- A. Type <tên kiểu xâu>: string;
- B. Type <tên kiểu xâu> = string;
- C. Type <tên kiểu xâu> = string[n];
- D. Type <tên kiểu xâu> := string;

II. Phần tự luận: (7 điểm)

Câu 1. Viết chương trình nhập vào một dãy số và in ra màn hình dãy sắp xếp tăng dần.

Câu 2. Viết chương trình tạo một file mới có tên là SOLIEU.DAT để thực hiện các công việc sau:

- + Lấy ngẫu nhiên một số nguyên dương bé hơn 100, nếu số đó khác không thì ghi vào tập tin sau đó lặp lại, nếu số đó bằng 0 thì dừng.
- + Đọc dãy số đang lưu trong tập tin và in ra màn hình.
- + Sắp xếp các số lưu trong các phần tử của tập tin sao cho khi đọc tuần tự từ tập tin thì ta được dãy số không giảm. Sau đó ghi dãy mới ra màn hình.

ĐÁP ÁN ĐỀ 10.

I. Phần trắc nghiệm: 1. A;

2. B

3.C

II. Phần tự luận:

Câu 1. (2 điểm)

Program SAPXEP;

Uses Crt;

```

Var A: array[1..20] of integer;
    i,j,tg,k,dem,n: integer;
Begin
    WriteLn('Nhap so phan tu cho mang');
    ReadLn(n);
    For i:=1 to n do
        Begin
            Write('A[' ,i,']=');
            ReadLn(a[i]);
        End;
    dem:=0;
    For i:=1 to n -1 do
        for j:=i+1 to n do
            if a[i] > a[j] then
                begin
                    tg:=a[i];
                    a[i]:=a[j];
                    a[j]:=tg;
                end;
    For i:=1 to n do
        WriteLn(A[i]);
    ReadLn;
End.

```

Câu 2. (5 điểm)

Program DAYSO;

Uses Crt;

Type fnguyen = File of word;

Var i,n:Integer;

PROCEDURE Taofile(st:string);

f:Fnguyen;

Rewrite(f);

Assign(f,st);

var f:fnguyen; n:word;

Begin

randomize;

Repeat

If n <> 0 then Write(f,n);

n:=random(100);

Until n=0;

Close(f);

End;

```

PROCEDURE Docfile(st:string);
  var f:Fnguyen; n:word;
  begin
    assign(f,st);
    reset(f);
    While not eof(f) do
      begin
        read(f,n);
        Writeln(n:2);
      end;
    Close(f);
  end;
PROCEDURE Xepthutu(st:string);
  var f:Fnguyen; n,i,j,a,b:word;
  Begin
    assign(f,st);
    n:=filesize(f);
    reset(f);
    for i:=0 to n-2 do
      begin
        seek(f,i);
        read(f,a);
        for j:= i+1 to n-1 do
          begin
            seek(f,j);
            read(f,b);
            if a> b then
              begin
                seek(f,i);
                seek(f,j);
                write(f,b);
                write(f,a);
              end;
            a:=b;
          end;
        end;
      Close(f);
    end;
  BEGIN
    CLRSCR;
    TAOFILE('SOLIEU.DAT');
    DOCFILE('SOLIEU.DAT');
  
```

```
WRITELN;  
XEPHUTU('SOLIEU.DAT');  
DOCFIL('SOLIEU.DAT');  
READLN;  
END.
```

ĐỀ 11.

I. Phần trắc nghiệm: (3 điểm)

1. Trong các phát biểu sau, phát biểu nào đúng?
 - A. Hai xâu bằng nhau là hai xâu có cùng cách khai báo dữ liệu.
 - B. Hai xâu khác nhau khi cách khai báo chúng khác nhau.
 - C. Hai xâu có độ dài bằng nhau khi tất cả các cặp kí tự đều giống nhau.
 - D. Hai xâu có cùng một kiểu luôn bằng nhau.
2. Cho St1:= 'abcd'; St2:= 'ABC'; khi đó st1+st2 cho kết quả nào?
 - A. abcd.
 - B. aabbcccd.
 - C. abcdABC.
 - D. ABCabcd.
3. Trong các phép toán sau phép nào đúng?
 - A. 'abcdh' > 'abcdef'.
 - B. 'abc' = 'ABC'.
 - C. 'cba' < 'abc'.
 - D. 'ccb' < 'abcdef'.

II. Phần tự luận: (7 điểm)

Câu 1: (2 điểm)

Viết chương trình nhập vào một mảng một chiều các số nguyên, sau đó thực hiện loại bỏ một phần tử ra khỏi mảng.

Câu 2: (5 điểm)

Để quản lí thi cuối khoá cho khối CD Tin Nhà trường lập danh sách sinh viên với các thông tin sau: số báo danh, họ tên, điểm Pascal, điểm cơ sở dữ liệu, điểm chính trị. Anh (chị) viết chương trình dưới dạng chương trình con thực hiện các công việc sau:

- a. Tạo một file có tên là SINHVIEN.DAT để lưu danh sách sinh viên và các thông tin về sinh viên. Yêu cầu việc nhập được lặp lại cho đến khi họ tên sinh viên được nhập vào là rỗng thì kết thúc.
- b. Hiện thị ra màn hình danh sách thí sinh đã dự thi cuối khoá với các thông tin số báo danh, tên, điểm 3 môn và tổng điểm 3 môn.

- c. Tìm và in ra danh sách tốt nghiệp với điều kiện không có môi thi nào dưới 5.

ĐÁP ÁN ĐỀ 11.

I. Phần trắc nghiệm: 1. C

2.C

3.A

II. Phần tự luận:

Câu 1. (2 điểm)

Program xoaphantu;

Uses Crt;

Var A: array[1..20] of integer;

i,k,x,n: integer;

Begin

Writeln('Nhap so phan tu cho mang');

Readln(n);

For i:=1 to n do

Begin

Write('A['i,']=');

Readln(a[i]);

End;

For i:=1 to n do

Begin

Writeln(A[i]);

End;

Writeln('Nhap vi tri can xoa');

Readln(k);

For i:= k+1 to n do

a[i-1]:=a[i];

Writeln('Ket qua sau khi xoa');

For i:=1 to n-1 do

Begin

Writeln(A[i]);

End;

Readln;

End.

Câu 2 (5 điểm)

Program QUANLYTHICUOIKHOA;

Uses Crt;

Type sinhvien = Record

Ten:string;

Sbd:Integer;

pas,cSDL,ctri:Integer;


```

        End;
        Fsv = File of sinhvien;
        Var i,n:Integer;
        f: Fsv;
PROCEDURE Taofile(st:string);
var f:fsv;sv:sinhvien;hoten:string;
Begin
    Assign(f,st);
    Rewrite(f);
    i:=1;
    Repeat
        Write('Nhap ho ten sinh vien thu', i,': ');
        Readln(hoten);
        If hoten <> ' ' then
            With sv do
                Begin
                    sbd:=i;
                    Ten:=hoten;
                    Write(' Nhap diem Pascal: '); Readln(pas);
                    Write(' Nhap diem csdl: '); Readln(csdl);
                    Write(' Nhap diem chinh tri: '); Readln(ctri);
                    Write(f,sv);
                    i:=+1;
                End
            Until hoten=' ';
        Close(f);
    End;
PROCEDURE Hienthidanhhsach(st:string);
var f:Fsv; sv:sinhvien;
begin
    assign(f,st);
    reset(f);
    While not eof(f) do
        begin
            read(f,sv);
            With sv do
                Writeln(sbd,ten,pas,csdl,ctri,pas+csdl+ctri);
            end;
        Close(f);
    end;
FUNCTION Kiem Booleansinhvien): Boolean;
Begin

```

```

With hs do
if (pas>5) and(csd>5)and(ctri>5) then
kiemtra:= True
Else kiemtra:=false;
End;
PROCEDURE Danhsachtrungtuyen(st:string);
var f:Fsv;sv :sinhvien;
Begin
    assign(f,st);
    reset(f);
    While not eof(f) do
    begin
        read(f,sv);
        if kiemtra(sv) then
            With sv do
                Writeln('Thi sinh co:',sbd,ten,pas,csdl,ctri);
            end;
        Close(f);
    end;
BEGIN
    CLRSCR;
    TAOFILE('SINHVIEN.DAT');
    HIENTHIDANHSACH('SINHVIEN.DAT');
    WRITELN('DANH SACH TOT NGHIEP');;
    DANHSACHTRUNG TUYEN('SINHVIEN.DAT');
    READLN;
END.

```

ĐỀ 12.

I. Phần trắc nghiệm: (3 điểm)

- Kết quả thực hiện của thủ tục str(123.456,st) là:
 - st = 123.456
 - st = 123
 - st = '123.456'
 - st = 123456
- Cho st = '123.456', khi thực hiện thủ tục str(st, a, k) cho kết quả:
 - a = 123.456.
 - a = 123456.
 - a = '123.456'
 - a = '123456'
- Đề khai báo biến kiểu bản ghi trực tiếp ta sử dụng cú pháp:
 - Var Tên biến = Record

Tên trường 1: Kiểu 1;
 Tên trường 2: Kiểu 2;

 Tên trường n: Kiểu n;
 End;

B. Var Tên biến : Record

Tên trường 1: Kiểu 1;
 Tên trường 2: Kiểu 2;

 Tên trường n: Kiểu n;
 End;

C. Var Tên biến : Tên kiểu ;

D. Var Tên biến : Record

Tên trường 1: Kiểu 1;
 Tên trường 2: Kiểu 2;

 Tên trường n: Kiểu n;

II. Phần tự luận: (7 điểm)

Câu 1. Viết thủ tục PHANTICH (n:Integer), để phân tích số nguyên n ra thừa số nguyên tố.

Ví dụ: $8=2*2*2$
 $24=2*2*2*3$

Câu 2. Viết chương trình nhập vào một số nguyên dương từ bàn phím. Tìm số đảo ngược của số đó rồi in kết quả ra màn hình theo cách Đề quy.

Ví dụ: nhập vào số 1243 in ra màn hình 3421

ĐÁP ÁN ĐỀ 12.

I. Phần trắc nghiệm: 1.C 2.B 3.B

II. Phần tự luận:

Câu 1. (2 điểm)

Viết thủ tục PHANTICH(n:Integer), để phân tích số nguyên n ra thừa số nguyên tố.

Ví dụ: $8=2*2*2$
 $24=2*2*2*3$

Uses crt;

Var n:Integer;

Procedure PHANTICH(n:Integer);

Var i:Integer;

```

Begin
  i:=2;
  m:=n;
  While m <> 1 Do
    Begin
      While (n MOD i=0) and (n div i <> 1) Do
        Begin
          Write(i:2, ' * ');
          n:=n Div i;
        End;
      m:=n Div i;
      i:=i+1;
    End;
  Write(i);
End;
Begin
  Write('Nhap n='); Readln(n);
  PHANTICH(n);
  Readln;
End.

```

Câu 2. (2 điểm)

```

Uses crt;
Var n:Integer;
Procedure DAOSO(n:Integer);
Begin
  Write(n MOD 10);
  if (n > 9) then
    DAOSO(n div 10);
End;
Begin
  Repeat
    Write('Nhap n='); Readln(n);
  Until (n>0)
  DAOSO(n);
  Readln;
End.

```

ĐỀ 13.

I. Phần trắc nghiệm: (3 điểm)

1. Sự khác nhau cơ bản về mặt lưu trữ trong bộ nhớ giữa kiểu dữ liệu tệp và các kiểu dữ liệu khác là:

A. Lưu trữ trong bộ nhớ RAM.

- B Kiểu tệp lưu trữ trong bộ nhớ ngoài (trên đĩa) còn các kiểu khác lưu ở bộ nhớ RAM.
- C Lưu trữ trong bộ nhớ ROM.
- D Không khác nhau.
2. Để khai báo biến tệp dạng văn bản ta sử dụng cú pháp nào?
- A Var Danh sách biến: File of;
- B Var Danh sách các biến : Text;
- C Var Danh sách biến: File;
- D Type Danh sách biến : Text;
3. Trong tập tin dạng văn bản thủ tục nào dưới đây được sử dụng để gán tên file cho biến?
- A Write(Biến file).
- B Rewrite(Biến file).
- C Assign(Biến file, tên file).
- D Assign(Biến file).

II. Phần tự luận: (7 điểm)

Người ta lưu thông tin các cán bộ trong cơ quan vào file có tên CANBO.DAT, mỗi cán bộ là một bản ghi gồm các trường: STT, Hoten, Ngaysinh, Diachi, HSLuong, HSPhucap. Hãy viết chương trình thực hiện các yêu cầu sau:

- Nhập danh sách cán bộ và lưu vào file, quá trình nhập dừng khi họ tên nhập vào là xâu rỗng và trường STT chương trình tự gán. (1 điểm)
 - In toàn bộ nội dung của file CANBO.DAT ra màn hình, nếu có, ngược lại thì thông báo "File không tồn tại". (1 điểm)
 - Sao chép thông tin các cán bộ có hệ số lương nằm trong khoảng từ x đến y, x và y là các số thực được nhập từ bàn phím vào một file khác. (1 điểm)
 - Thông tin về hệ số lương của cán bộ có họ tên là Bhoten, ngày sinh là Bngaysinh và địa chỉ là Bdiachi bị sai lệch. Hãy sửa hệ số lương của cán bộ này với dữ liệu nhập từ bàn phím. (1 điểm)
 - Sao chép nội dung của file CANBO.DAT vào file văn bản CANBO.TXT, mỗi cán bộ tương ứng một dòng. (1 điểm)
- Chương trình chính (1 điểm)

ĐÁP ÁN ĐỀ 13.

I. Phần trắc nghiệm: 1. B. 2.B 3.C

II. Phần tự luận:

Program QLCB;

Type

St20 = String[20];

St10 = String[10];

Canbc = Record


```

STT:integer;
Hoten, Diachi: St20,
Ngaysinh: St10; {dd/mm/yyyy}
HSluong, HSPhucap: real;
end;
Var
  f: File of Canbo;
  g:Text;
  St: String;
  filename:String;
  Cb:canbo;
  Bhoten:st20;
  i:word;
Procedure NhapDL();
Begin
  assign(f,'CANBO.DAT');
  rewrite(f);
  i:=1;
  repeat
    writeln('Nhap thong tin cua cac can bo');
    writeln('Thong tin cua can bo thu ', i);
    write('Ho ten: ');
    readln(Bhoten);
    if Bhoten <> " then
      begin
        Cb.STT:=i;
        Cb.hoten:= Bhoten;
        write('Ngay sinh (dd/mm/yyyy): ');
        readln(Cb.ngaysinh);
        write('Dia chi: ');
        readln(Cb.Diachi);
        write('He so luong: ');
        readln(Cb.HSluong);
        write('He so phu cap: ');
        readln(Cb.HSpheucap);
        write(f,Cb);
      end;
    inc(i);
  until Bhoten = "";

```

```

    close(f);
end;

Procedure InDL();
Begin
    assign(f,'Canbo.dat');
    {$I-}
    reset(f);
    {$I+}
    if IOResult <> 0 then
        Begin
            writeln('File khong ton tai');
            exit;
        End;
    writeln(#32:10, 'DANH SACH CAN BO');
    writeln('STT',#32:6,'HO TEN',#32:8,'NGAY SINH',#32:4,'DIA
CHI',#32:4, 'HSluong', #32:4, 'HSphucap' );
    while not eof(f) do
        begin
            read(f,Cb);
            with Cb do

writeln(STT:3,hoten,#32:20,length(hoten),ngaysinh,#32:2,Diachi,#32:10-
length(Diachi),HSluong:5:2,HSphucap:5:2);
        end;
    close f);
    readln;
End;

```

```

Procedure SuaDL();
var
    Bngaysinh,Bdiachi:St10;
Begin
    assign(f,'Canbo.dat');
    {$I-}
    reset(f);
    {$I+}
    if IOResult<>0 then
        begin

```

```

        writeln('File khong ton tai');
        exit;
    end;
write('Ho ten can bo: ');
readln(bhoten);
write('Ngay sinh: ');
readln(Bngaysinh);
write('Dia chi: ');
readln(bdiachi);
while not Eof(f) do
    begin
        read(f,Cb);
        with Cb do
            if (hoten=bhoten) and ((ngaysinh=bngaysinh) and (diachi=bdiachi)) then
                begin
                    write('Nhap HSluong can sua: ');
                    readln(HSluong);
                    n:=filepos(f);
                    seek(f,n-1);
                    write(f,Cb);
                    exit;
                end;
    end;
    Close(f);
readln;
End;
Procedure SaoChepDL();
Begin
    assign(f,'Canbo.dat');
    {$I-}
    reset(f);
    {$I+}
    if IOResult <> 0 then
        Begin
            writeln('File khong ton tai');
            exit;
        End;
    assign(f1,'CANBOXY.DAT');
    rewrite(f1);

```

```

repeat
    write('Nhap khoan he so luong tu x den y: ');
    readln(x,y);
until (x<y)
    writeln(#32:10, 'DANH SACH CAN BO HSL TU ', x,' TOI ',y);
    writeln('STT',#32:6,'HO TEN',#32:8,'NGAY SINH',#32:4,'DIA
CHI',#32:4, 'HSluong', #32:4, 'HSphucap' );
    while not eof(f) do
        begin
            read(f,Cb);
            if (Cb.HSluong>=x) and (Cb.HSluong <=y) then
                write(f1,Cb)
            end;
        close(f);
        readln;
    End;

```

Procedure SCDFFileText();

Begin

```

    assign(f,'Canbo.dat');
    {$I-}
    reset(f);
    {$I+}
    if IOResult <>0 then
        begin
            writeln('File khong ton tai');
            exit;
        end;
    rewrite(g);
    while not Eof(f) do
        begin
            read(f, Cb);
            with Ct do
                begin
                    ST:= Str(STT) + hoten + #32 + ngaysinh + #32 + Diachi + #32 +
Str(HSluong) + #32 + Str(HSphucap);
                    writeln(g,St);
                end;
        end;

```

```

        end;
    Close(f);
    Close(g);
    readln;
End;

BEGIN
Repeat
    Clrscr;
    Writeln('CHUONG TRINH QUAN LY CAN BO');
    Writeln('1. Nhap danh sach can bo');
    Writeln('2. In danh sach can bo');
    Writeln('3. Sao chép danh sach can bo co HSL tu x den y');
    Writeln('4. Sua He so luong can bo');
    Writeln('5. Sao chép Du lieu thanh File text');
    Writeln('<ESC>: Thoat');
    c:=Readkey;
    Case c Of
        '1': NhapDL();
        '2': InDL();
        '3': SaochepDL();
        '4': SuaDL;
        '5': SCDLFileTex();
    End;
Until c=#27;
END.

```

ĐỀ 14.

I. Phần trắc nghiệm: (3 điểm)

1. Biến địa phương là:
 - A. Biến được khai báo sau từ khoá Var của chương trình con.
 - B. Biến được khai báo sau từ khoá Var của chương trình chính.
 - C. Biến được khai báo trong chương trình con hàm và thủ tục
 - D. Đáp án a và c đều đúng.
2. Phát biểu nào dưới đây là sai ?
 - A. Một thủ tục có thể không có tham số hình thức nào.
 - B. Một thủ tục có thể có một hoặc nhiều tham số hình thức.
 - C. Một thủ tục bắt buộc chỉ có một tham số hình thức.

D. Tất cả đều đúng.

3. Trong chương trình khi thực hiện lệnh Tên hàm: = Biểu thức; thì:

A. Kiểu giá trị của biểu thức có thể nhận bất cứ kiểu gì.

B. Kiểu của biểu thức phải trùng với kiểu của giá trị trả về của hàm.

C. Kiểu của biểu thức là kiểu bản ghi.

D. Kiểu của biểu thức là kiểu số.

II. Phần tự luận: (7 điểm)

Câu 1. Viết chương trình in ra màn hình tam giác Pascal. Ví dụ, với $n=4$ sẽ in ra hình sau:

```
1
1   1
1   2   1
1   3   3   1
1   4   6   4   1
```

Câu 2. Viết chương trình nhập vào một chuỗi ký tự từ bàn phím. Tìm chuỗi đảo ngược của chuỗi đó rồi in kết quả ra màn hình theo cách Đề quy.

ví dụ: nhập vào chuỗi "chao ban!" thì in ra màn hình chuỗi "!nab oahc"

ĐÁP ÁN ĐỀ 14.

I. Phần trắc nghiệm: 1.D 2.C 3.B

II. Phần tự luận:

Câu 1.

Uses Crt;

Var Dong:Array[0..20] Of Byte;

 n,i,j:Byte;

Begin

 Write(n= '); Readln(n);

 Clrscr;

 Dong[0]:=1;

 Writeh(Dong[0]:4);

 {Khoitao gia tri cua dong}

 For i:=1 To n Do Dong[i]:=0;

 {Voi moi dong i}

 For i:=1 To n Do

 Begin

 For j=i DownTo 1 Do

 Begn

```

    Dong[j]:=Dong[j-1]+Dong[j];
    Write(Dong[j]:4);
End;
Writeln(Dong[i]:4);
End;
Readln;
End.

```

Câu 2.

```

Function DeQuy(St:String):String;
Begin
    If Length(St)<=1 Then DeQuy:=St
    Else DeQuy:=St[Length(St)] + DeQuy(Copy(St,1,Length(St)-1))
End;
Begin
    Write('Nhap xau St: '); Readln(St);
    Write('Xau dao nguoc: ', XauDao(St));
    Readln;
End.

```

ĐỀ 15.

I. Phần trắc nghiệm: (3 điểm)

1. Phát biểu nào dưới đây là đúng ?
 - A. Chương trình con là hàm có thể gọi chương trình con thủ tục.
 - B. Chương trình con thủ tục có thể gọi chương trình con hàm.
 - C. Một chương trình con có thể có lời gọi các chương trình con khác đã được định nghĩa trước nó.
 - D. Cả ba phương án đều đúng.
2. Phát biểu nào dưới đây là đúng?
 - A. Thời gian tồn tại của biến cục bộ là từ khi chương trình con đó thực hiện cho đến khi kết thúc.
 - B. Thời gian tồn tại của biến toàn cục là toàn bộ thời gian chương trình chính hoạt động.
 - C. Thời gian hoạt động của biến cục bộ và toàn cục là như nhau.
 - D. Đáp án a và b đều đúng.
3. Phát biểu nào dưới đây là sai?

- A. Tên của các tham số hình thức và tên của các biến địa phương trong cùng một chương trình con không được trùng nhau.
- B. Tên của các biến địa phương có thể trùng tên với tên biến toàn cục.
- C. Biến toàn cục có thể tham gia trong các chương trình con của nó.
- D. Biến địa phương có thể tham gia trong bất cứ chương trình con nào và cả chương trình chính.

II. Phần tự luận: (7 điểm)

Để quản lý sinh viên, người quản lý tạo một file SINHVIEN.DAT để lưu thông tin của một lớp sinh viên. Mỗi sinh viên cần những thông tin sau: Họ tên, Ngày sinh, Quê quán, Điểm trung bình, Xếp loại (trường xếp loại do chương trình tự tính lấy dựa vào điểm trung bình như sau: nếu điểm trung bình < 5 thì xếp loại 'D', nếu $5 \leq$ điểm trung bình < 6.5 thì xếp loại 'C', nếu $6.5 \leq$ điểm trung bình < 8 thì xếp loại 'B', trường hợp còn lại xếp loại 'A'). Anh (chị) hãy viết các thủ tục theo các yêu cầu sau để giúp người quản lý:

- a. Nhập sinh viên của một lớp vào file SINHVIEN.DAT, quá trình nhập dừng khi họ tên nhập vào là xâu rỗng (1 điểm)
 - b. In toàn bộ nội dung của file SINHVIEN.DAT ra màn hình. nếu có, ngược lại thì thông báo "File không tồn tại". (1 điểm)
 - c. In danh sách tất cả sinh viên có thông tin lưu trong file SINHVIEN.DAT xếp loại khá ('B') trở lên. (1 điểm)
 - d. Thông tin về điểm của sinh viên có họ tên là Bhoten, ngày sinh là Bngaysinh và quê quán là Bquequan bị sai lệch. Hãy sửa điểm và xếp loại của sinh viên này với dữ liệu nhập từ bàn phím. (1 điểm)
 - e. Sao chép nội dung của file SINHVIEN.DAT vào file văn bản SINHVIEN.TXT sao cho mỗi sinh viên lưu trong một dòng. (1 điểm)
- Chương trình chính (1 điểm)

ĐÁP ÁN ĐỀ 15.

I. Phần trắc nghiệm: 1.C 2.B 3.D

II. Phần tự luận:

Program QLSV;

Type

St20 = String[20];

St10 = String[10];

SinhVien = record

Hoten: St20;

```

        Ngaysinh,Quequan: St10;
        DiemTb: real;
        Xeploai: Char;
    end;
Var
    f: File of SinhVien;
    g:Text;
    St: String;
    filename:String;
    Sv: sinhvien;
    Bhoten:st20;
    i:word;
Procedure NhapDL();
Begin
    assign(f,'SINHVIEN.DAT');
    rewrite(f);
    i:=1;
    repeat
        writeln('Nhap thong tin cua cac sinh vien');
        writeln('Thong tin cua sinh vien thu ', i);
        write('Ho ten: ');
        readln(Bhoten);
        if Bhoten <> " then
            begin
                sv.hoten:= Bhoten;
                write('Ngay sinh (dd/mm/yyyy): ');
                readln(sv.ngaysinh);
                write('Quequan: ');
                readln(sv.quequan);
                write('Diem trung binh: ');
                readln(sv.diemtb);
                if sv.diemtb<5 then
                    sv.xeploai:='D'
                else
                    if sv.diemtb<6.5 then
                        sv.xeploai:='C'
                    else
                        if sv.diemtb<8 then
                            sv.xeploai:='B'

```

```

        else
            sv.xeploai:='A';
        write(f,sv);
    end;
    inc(i);
    until Bhoten = "";
    close(f);
end;
Procedure InDL();
Begin
    assign(f,'Sinhvien.dat');
    {$I-}
    reset(f);
    {$I+}
    if IOResult <> 0 then
        Begin
            writeln('File khong ton tai');
            exit;
        End;
    writeln(#32:10, 'DANH SACH SINH VIEN');
    writeln(#32:6,'HO TEN',#32:8,'NGAY SINH',#32:4,'QUE QUAN
DTB');
    while not eof(f) do
        begin
            read(f,sv);
            with sv do

writeln(hoten,#32:20,length(hoten),ngaysinh,#32:2,quequan,#32:10-
length(quequan),Dientb:5:2);
            end;
        close(f);
        readn;
    End;
Procedure InDLK();
Begin
    assign(f,'sinhvien.dat');
    {$I-}
    reset(f);
    {$I+}

```



```

if IOResult <> 0 then
begin
    writeln('File không tồn tại');
    exit;
end;
n:=0;
writeln('Danh sách sinh viên đã loại khá trở lên');
while not Eof(f) do
begin
    read(f,sv);
    with sv do
        if xeploai <= 'B' then { (xeploai = 'B') or (xeploai = 'A') }
        begin
            writeln(hoten,ngaysinh,quequan,diem:b);
            inc(n);
        end;
    end;
    close(f);
writeln('Danh sách này gồm ',n,' sinh viên');
readln;
end;
Procedure SuaDL();
var
    Bngaysinh,Bquequan:St10;
Begin
    assign(f,'sinhvien.dat');
    {$I-}
    reset(f);
    {$I+}
    if IOResult <> 0 then
    begin
        writeln('File không tồn tại');
        exit;
    end;
    write('Họ tên sinh viên: ');
    readln(bhoten);
    write('Ngày sinh: ');
    readln(Bngaysinh);
    write('Que quán: ');

```

```

readln(bquequan);
while not Eof(f) do
    begin
        read(f,sv);
        with sv do
            if (hoten=bhoten) and ((ngaysinh=bngaysinh) and (quequan=bquequan))
            then
                begin
                    write('Nhap dtb can sua: ');
                    readln(diemtb);
                    if diemtb <5 then
                        xeploai:='D'
                    else
                        if diemtb <6.5 then
                            xeploai:='C'
                        else
                            if diemtb <8 then
                                xeploai:='B'
                            else
                                xeploai:='A';
                    n:=filepos(f);
                    seek(f,n-1);
                    write(f,sv);
                    exit;
                end;
            end;
        Close(f);
    readln;
End;

```

Procedure SaochepDL();

Begin

 assign(f,'sinhvien.dat');

 {\$I-}

 reset(f);

 {\$I+}

 if IOResult <>0 then

 begin

 writeln('File khong ton tai');

```

        exit;
    end;
    rewrite(g);
    while not Eof(f) do
        begin
            read(f, Sv);
            with Sv do
                begin
                    Str(diemtb,bdiem:5:2);
                    St:= hoten+#32+ngaysinh+#32+quequan+#32+Bdiem;
                    writeln(g,St);
                end;
            end;
        end;
        Close(f);
        Close(g);
        readln;
    End;

```

BEGIN

Repeat

Clrscr;

Writeln('CHUONG TRINH QUAN LY SINH VIEN');

Writeln('1. Nhap danh sach sinh vien');

Writeln('2. In danh sach sinh vien');

Writeln('3. In danh sach sinh vien loai kha tro len');

Writeln('4. Sua diem sinh vien');

Writeln('5. Sao chep');

Writeln('<ESC>: Thoat');

c:=Readkey;

Case c Of

'1': NhapDL();

'2': InDL();

'3': InDLK();

'4': SuaDL;

'5': SaochepDL();

End;

Until c=#27;

END.

D. MỘT SỐ BÀI TẬP NÂNG CAO

Bài 1.

Số kì lạ là những số có các tính chất: Chia hết cho 2, chia hết cho 3 nhưng không chia hết cho 5.

Hãy lập chương trình để kiểm tra một số được nhập vào từ bàn phím có phải là một số kì lạ hay không? In thông báo kết quả lên màn hình.

Bài 2.

Hiển thị 1000 số nguyên được tạo một cách ngẫu nhiên (trong phạm vi 10000) lên màn hình với 10 số / 1 dòng, mỗi số cách nhau 5 khoảng trắng.

Bài 3.

Giải bài toán cổ: “Trăm trâu trăm cỏ, trâu đứng ăn năm, trâu nằm ăn ba, lụ khụ trâu già, ba con một bó”. Hỏi có bao nhiêu trâu đứng, bao nhiêu trâu nằm và bao nhiêu trâu già.

Bài 4.

Số hoàn hảo là những số có tính chất “Tổng tất cả các ước số của nó (nhỏ hơn số đó - kể cả 1) đúng bằng chính nó”. Ví dụ $6 = 1+2+3$ là một số hoàn hảo. Tìm tất cả các số hoàn hảo nằm trong phạm vi từ 1 đến 10000.

Bài 5.

Lập chương trình nhập vào một dãy n số nguyên, lưu vào mảng A và thực hiện một số công việc sau:

- In kết quả nhập với 5 số / dòng, các số cách nhau 3 khoảng trắng.
- Sắp xếp mảng A theo thứ tự tăng dần và in kết quả sau khi sắp xếp.
- Tính tổng tất cả các số chẵn có trong mảng.
- Tính tổng tất cả các SHH có trong mảng.
- Tính tích của tất cả các số nguyên tố có trong mảng.
- Nhập vào một số bất kì, kiểm tra xem số đó đã tồn tại trong mảng hay chưa, nếu chưa tồn tại thì chèn số đó vào mảng sao cho vẫn giữ được trật tự sắp xếp của mảng. Nếu số đó đã tồn tại thì thực hiện xóa phần tử đó khỏi mảng.

Bài 6.

Lập chương trình nhập vào một dãy kí tự. Thực hiện một số công việc sau:

- In dãy kí tự vừa nhập ra màn hình.
- Đếm số kí tự Alphabet có trong dãy.
- Đổi thành kí tự hoa những kí tự thuộc Alphabet.
- Tính tổng của tất cả các kí tự là chữ số (Biết kí tự 0 có số thứ tự trong

mảng mã ASCII là 48).

- Sắp xếp dãy kí tự trên và in kết quả sắp xếp.

Bài 7.

Trộn hai mảng số nguyên A, B đã được sắp xếp tăng dần thành một mảng C cũng có trật tự sắp xếp tăng dần.

Bài 8.

Dùng hàm Random để gieo ngẫu nhiên 100 số nguyên dương trong phạm vi từ 100 đến 10000, lưu vào mảng A.

- In dãy số vừa tạo.
- Liệt kê tất cả các số nguyên tố có trong mảng A.
- Đếm xem có bao nhiêu phần tử là số hoàn hảo.

Bài 9.

Viết chương trình nhập từ bàn phím họ và tên của một người, sau đó in ra phần tên của người đó. Ví dụ: “Tran Thanh Hai” thì in ra “Hai”.

Bài 10.

Lập chương trình để lưu tên (không có phần họ và chữ lót) của 10 người được nhập vào từ bàn phím, in ra danh sách vừa nhập.

- Sắp xếp tên theo thứ tự tăng dần và hiển thị kết quả sau khi đã sắp xếp.
- Kiểm tra và sửa lại tên của 10 người trên nếu chữ cái đầu không được viết hoa hoặc chữ cái sau không được viết hoa.

Bài 11.

Lập chương trình tạo một file số nguyên để lưu n số nguyên bất kì vào file và thực hiện một số công việc sau:

- Đọc phần tử thứ k có trong file.
- Kiểm tra xem một số nguyên x được nhập vào từ bàn phím đã có trong file hay chưa. Nếu chưa có thì bổ sung số đó vào file.
- Đọc file và hiển thị ra màn hình.

Bài 12.

Giả sử trong thư mục C:\TP đã có file văn bản VB.TXT. Hãy lập chương trình thực hiện các công việc sau:

- Đếm số kí tự có trong file.
- Đếm số kí tự theo từng loại kí tự có trong file.
- Đếm số từ có trong file.

E. MỘT SỐ ĐỀ THI OLYMPIC

Bài 1

Cho n số nguyên dương $A_1, A_2, \dots, A_n$ ($n \leq 100$ và $A_i \leq 200$ với $i=1, \dots, n$). Hãy tìm cách chia n số đã cho thành 2 nhóm sao cho tổng các số trong mỗi nhóm chênh lệch nhau ít nhất.

Yêu cầu:

Dữ liệu vào cho bởi file BAI1.INP

- Dòng thứ nhất ghi số n ;
- Các dòng tiếp theo ghi giá trị của các A_i ($i=1, \dots, n$).

Kết quả chạy chương trình cho hiển thị lên màn hình:

- Phần đầu in giá trị chênh lệch giữa hai nhóm;
- Phần thứ hai in các nhóm thuộc nhóm 1;
- Phần thứ hai in các nhóm thuộc nhóm 2.

Ví dụ

Test1:

Dữ liệu vào:

```
5
1      2      5
10     50
```

Kết quả chạy chương trình:

```
Chênh lệch giữa hai nhóm là: 32
Nhóm 1: 10      5      2      1
Nhóm 2: 50
```

Test2:

Dữ liệu vào:

```
10
197  74  67
23   58  127
75   93  163  61
```

Kết quả:

```
Chênh lệch giữa hai nhóm là: 4
Nhóm 1: 61  163  93  127  23
Nhóm 2: 197  74  67  58  75
```

Bài 2

Nhập từ bàn phím một số nguyên n ($4 < n < 100000$). Chọn nhiều nhất m số nguyên tố khác nhau, sao cho tổng của m số nguyên tố này nhỏ hơn hoặc bằng n .

Yêu cầu:

Đưa ra màn hình: số m lớn nhất và có bao nhiêu cách chọn với m số nguyên tố (gọi d là số cách chọn). Không yêu cầu liệt kê các cách chọn.

Ví dụ:

Test1

Input: $n=8$

Output: Đưa ra màn hình: $m=2, d=3$

($2+3 < 8; 2+5 < 8; 3+5 \leq 8$)

Bài 3: Bài toán trình tự gia công ngắn nhất

Trong một phân xưởng sản xuất phụ tùng xe gắn máy có trang bị hai máy A và B dùng để sản xuất. Biết rằng, để sản xuất 1 phụ tùng, nó cần được gia công lần lượt trên máy A rồi đến máy B.

Hãy lập trình xây dựng trình tự gia công n phụ tùng trên hai máy sao cho thời gian gia công là ngắn nhất.

Yêu cầu:

File Test: *GIACONG.INP*

Dòng đầu: Số n chỉ số phụ tùng cần gia công.

Dòng thứ hai: Các số T_i ($i=1,2,...,n$) chỉ thời gian gia công phụ tùng thứ i trên máy A, mỗi số cách nhau ít nhất một khoảng trống.

Dòng thứ ba: các số T_i ($i=1,2,...,n$) chỉ thời gian gia công phụ tùng thứ i trên máy B, mỗi số cách nhau ít nhất một khoảng trống.

Xuất ra:

Trình tự gia công các phụ tùng.

Ví dụ:

Test1:

File: *GIACONG1.INP*

4

1 5 3 4

2 3 2 5

Kết quả: Trình tự gia công: 1 4 2 3

Test2:

File: *GIACONG2.INP*

5

6 7 8 5 4

9 8 3 2 1

Kết quả: Trình tự gia công: 1 2 3 4 5.

Bài 4 Nhận dạng hằng số thực

Người ta định nghĩa, một hằng số được coi là số thực nếu ngoài các chữ số ra nó còn có dấu (.) thì mỗi bên của dấu chấm phải có ít nhất một chữ số. Ngoài ra, ở trước số và trong số mũ còn có thể có một dấu cộng (+) hoặc trừ (-). Phần số lũy thừa phải là số nguyên. Số thực có thể có các khoảng trống ở trước hoặc sau nhưng bên trong thì không. Hằng số thực không có giới hạn về giá trị.

Viết chương trình kiểm tra xem các chuỗi kí tự cho trước có phải là những số thực theo định nghĩa trên đây hay không.

Yêu cầu:

Dữ liệu vào:

Dữ liệu vào cho trong tập tin văn bản: SOTHUC.INP gồm nhiều dòng cho nhiều trường hợp cần kiểm tra. Dòng cuối cùng chứa kí tự $\square^*\square$, dòng này không được xét trong chương trình. Mỗi dòng tối đa 255 kí tự.

Dữ liệu ra:

Kết quả ghi trong tập tin văn bản SOTHUC.OUT gồm nhiều dòng. Mỗi dòng chứa câu trả lời là “Đúng” hoặc “Sai” tương ứng với mỗi dòng trong tập tin SOTHUC.INP.

Ví dụ:

Test 1:

Dữ liệu vào:	Dữ liệu ra
1,2	DUNG
5,3000	DUNG
1.	SAI
1.0E55	DUNG
+1E-12	DUNG
E-14	SAI
E	SAI
*	

Bài 5: Bài toán xâu trong cực đại

Xâu S gọi là xâu trong của T nếu S nhận được từ T bằng cách xoá đi một số kí tự nào đó.

Ví dụ: “ABC” là xâu trong của “MAXBMXCA”.

Bài toán:

Cho hai xâu T1 và T2. Khi đó hãy tìm một xâu S là xâu trong chung của T1 và T2 có độ dài cực đại.

Test

Input:

T1= " AMXAL"

T2= " BXAMLA"

Output:

S= "XAL"; "AML"; "AMA"

Bài 6: Bài toán Đội cờ

Có hai đội cờ vua A và B thi đấu với nhau. Mỗi đội cử ra n kì thủ, mỗi kì thủ của đội B chỉ đấu một trận và chỉ đấu với một kì thủ của đội A và ngược lại. Vậy có tất cả n trận đấu. Đội nào thắng được 2 điểm, hoà 1 điểm, thua 0 điểm. Cho đội B được chọn cặp thi đấu.

Yêu cầu:

Lập chương trình để đội B chọn được các cặp thi đấu sao cho tổng số điểm của đội B là cao nhất. Cho biết trình độ của kì thủ thứ i của hai đội A và B lần lượt là a[i] và b[i] ($i=1,...,n$) và giả sử trong thi đấu hai kì thủ có trình độ bằng nhau sẽ hoà và kì thủ nào có trình độ cao hơn sẽ thắng.

Dữ liệu vào:

Ghi trên file văn bản covua.in, gồm n + 1 dòng:

- Dòng đầu ghi số nguyên dương n, $1 \leq n \leq 1000$
- Trên dòng thứ i + 1 ($1 \leq i \leq n$) ghi hai số nguyên a[i] và b[i] ($1 \leq a[i], b[i] \leq 100$) và cách nhau một khoảng trắng.

Dữ liệu ra:

Ghi trên file văn bản covua.out, gồm n+1 dòng:

- Dòng đầu là số nguyên T là tổng số điểm cao nhất mà đội B có thể đạt được.
- Trên dòng thứ i+1 ($1 \leq i \leq n$) là số nguyên dương x[i] ($1 \leq i \leq n$), trong đó x[i] là số thứ tự của kì thủ của đội B phải đấu với kì thủ thứ i của đội A để tổng số điểm của đội B đạt được là T.

Test

Covua.in

covua.out

4

5

7 8

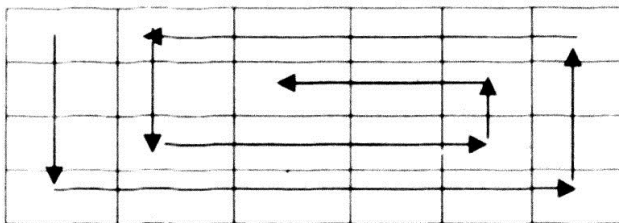
1

5 6

2

Bài 7

Một ma trận số nguyên cấp $m \times n$ được gọi là ma trận xoáy giảm nếu như duyệt ma trận theo đường mũi tên ở hình sau thì ta được một dãy giảm dần.



Cho trước một ma trận $A = (a_{i,j})_{m,n}$. Kiểm tra xem A có phải là ma trận xoáy giảm hay không? Nếu không phải thì hãy sắp xếp lại để được một ma trận xoáy giảm.

Dữ liệu vào ghi trong file MT.INP gồm:

- Dòng đầu ghi hai giá trị: m, n ($m, n \leq 50$)
- m dòng tiếp theo lần lượt lưu giá trị m hàng của ma trận A . Các số khác nhau trong một hàng cách nhau ít nhất một khoảng trắng.

Dữ liệu ra ghi trong file MT.OUT, trong đó:

- Nếu A không phải là ma trận xoáy giảm thì:
 - o Dòng đầu tiên ghi số 0
 - o m dòng tiếp theo ghi các giá trị của m hàng của ma trận A sau khi đã sắp xếp.
- Nếu A là ma trận xoáy giảm thì chỉ ghi một số 1 lên file

Bài 8

Một lớp có $m \times n$ chỗ ngồi gồm m hàng ghế, mỗi hàng ghế có n học sinh. Để chuẩn bị cho kì thi học sinh giỏi, ban cán sự lớp nghĩ ra sáng kiến: Mỗi người trong ban cán sự sáng tác ra 1 đề, sau đó sao thành một số bản để đưa cho người bên cạnh:

- Bên phải, bên trái, trên và dưới
- Mỗi người đúng một bản

Số người được đưa có thể là 2, 3, 4

Sau đó tất cả học sinh trong lớp phải thông báo số đề mình nhận được

Lập trình xác định vị trí ngồi của các bạn trong ban cán sự lớp (ghi ra tất cả các cách có thể có), yêu cầu kiểm tra dữ liệu nhập.

Dữ liệu vào ghi trong file VITRI.INP có dạng:

- Dòng thứ nhất ghi 2 số m và n
- n dòng tiếp theo, mỗi dòng ghi n số cách nhau một khoảng trắng

Dữ liệu ra:

Xuất ra màn hình vị trí ngồi của các bạn trong ban cán sự lớp.

Test1

VITRI.INP

Màn hình

4 6

0 1 0 1 1 0

0 0 0 0 0 0

1 0 3 1 1 1

0 * 0 * * 0

0 2 0 2 1 0

0 0 * 0 0 0

0 0 1 0 0 0

0 0 0 0 0 0

Test2

VITRI.INP

Màn hình

3 3

0 1 0

* 0 0 0 0 * 0 0 0

1 0 1

0 0 0 0 0 0 0 * 0

0 1 0

0 0 * * 0 0 0 0 0

Test3

VITRI.INP

Màn hình

4 4

0 1 0 0

Không có cách sắp xếp, dữ liệu vào sai.

1 1 1 0

0 1 0 0

0 0 0 0

Bài 9

Cho một file chứa các số nguyên dương có thể trùng nhau theo cấu trúc sau:

- Dòng đầu tiên chứa n là số lượng các số nguyên dương
- Dòng tiếp theo lần lượt chứa n số nguyên dương
- Mỗi số cách nhau một khoảng trắng

Hãy chọn từ đó ra một tập nhỏ nhất các số nguyên dương sao cho mọi số trong tập tin đã cho đều có thể viết dưới dạng tích của một, một vài hay tất cả các số trong tập đã chọn, mỗi số có thể dùng nhiều lần trong phép nhân nếu cần. Kết quả đưa ra có dạng:

- Dòng đầu tiên là số lượng số nguyên dương được chọn
- Dòng tiếp theo là các số nguyên dương đã chọn

Test1: Vào:

12

15 60 5 2 200 3 2 40 15 24 5
3

Ra:

3

2 3 5

Test2: Vào:

16

2 45 4 51 15 34 200 40 17 3 5 9 18 50 120 6

Ra:

4

2 3 5 17

Bài 10

Một loại động vật quý hiếm cần được di chuyển từ địa điểm A qua B. Loài vật này nhạy cảm với nhiệt độ nên dễ mắc bệnh. Nếu nhiệt độ chênh lệch giữa hai địa điểm đi qua liên tiếp càng cao thì nguy cơ mắc bệnh càng lớn. Hãy tìm một hành trình từ A đến B sao cho nguy cơ mắc bệnh của loài vật là thấp nhất. Cho A là điểm đầu và B là điểm cuối. Hành trình từ A đến B có thể đi qua một vài hay tất cả các điểm đã cho trong file.

Dữ liệu vào có:

- Dòng đầu chứa n là số địa điểm
- Dòng tiếp theo chứa n số nguyên là nhiệt độ từng điểm
- Mỗi số cách nhau một khoảng trắng

Kết quả ra:

- Dòng đầu là nhiệt độ chênh lệch lớn nhất giữa hai điểm đi qua liên tiếp trong màn hình.
- Dòng thứ hai lần lượt cho biết nhiệt độ các điểm thuộc hành trình từ A đến B tìm được.

Test: Vào:

8

16 21 22 28 18 2 14 19

Ra:

2

16 8 19

Test2.

Vào:

18

43 40 17 55 49 64 58 68 25 82 14 13 81 101 6 78 90 72

Ra:

6

43 49 55 58 64 68 72

MỤC LỤC

Chương 1

MỘT SỐ KHÁI NIỆM VỀ LẬP TRÌNH VÀ NGÔN NGỮ LẬP TRÌNH

Bài 1. Khái niệm lập trình và ngôn ngữ lập trình	11
Bài 2. Các thành phần của ngôn ngữ lập trình	15

Chương 2

CHƯƠNG TRÌNH ĐƠN GIẢN

Bài 3. Cấu trúc chương trình.....	19
Bài 4. Một số kiểu dữ liệu chuẩn	23
Bài 5. Khai báo biến	27
Bài 6. Phép toán, biểu thức, câu lệnh gán	30
Bài 7. Các thủ tục chuẩn vào / ra đơn giản.....	35
Bài 8. Soạn thảo, dịch, thực hiện và hiệu chỉnh chương trình.....	41

Chương 3

CẤU TRÚC RỄ NHÁNH VÀ LẶP

Bài 9. Cấu trúc rẽ nhánh	49
Bài 10. Cấu trúc lặp	57

Chương 4

Kiểu dữ liệu có cấu trúc

Bài 11. Kiểu mảng	70
Bài 12. Kiểu xâu	87
Bài 13. Kiểu bản ghi (record).....	98

Chương 5

TỆP VÀ THAO TÁC VỚI TỆP

Bài 14. Kiểu dữ liệu tệp	113
Bài 15. Thao tác với tệp	115
Bài 16. Ví dụ làm việc với tệp.....	124

Chương 6

CHƯƠNG TRÌNH CON VÀ LẬP TRÌNH CÓ CẤU TRÚC

Bài 17. Chương trình con và phân loại	135
Bài 18. Ví dụ về cách viết và sử dụng chương trình con.....	138
Bài 19. Thư viện chương trình con chuẩn	150
ĐÁP ÁN VÀ HƯỚNG DẪN GIẢI	152

Chương 7

MỘT SỐ BÀI TOÁN NÂNG CAO, CÁC BÀI THỰC HÀNH DÙNG TRÊN MÁY VÀ MỘT SỐ ĐỀ KIỂM TRA DÙNG THAM KHẢO

A. Kiến thức mở rộng: Đồ họa và âm Thanh	176
B. Các bài thực hành mẫu	182
C. Một số đề kiểm tra tham khảo	195